


Summer 10-1-2018

Implementation of Secure DNP3 Architecture of SCADA System for Smart Grids

Uday Bhaskar Boyanapalli

Follow this and additional works at: https://digitalcommons.kennesaw.edu/cs_etd

 Part of the [Computer and Systems Architecture Commons](#), [Digital Communications and Networking Commons](#), [Information Security Commons](#), [Power and Energy Commons](#), [Software Engineering Commons](#), and the [Systems Architecture Commons](#)

Recommended Citation

Boyanapalli, Uday Bhaskar, "Implementation of Secure DNP3 Architecture of SCADA System for Smart Grids" (2018). *Master of Science in Computer Science Theses*. 17.
https://digitalcommons.kennesaw.edu/cs_etd/17

This Thesis is brought to you for free and open access by the Department of Computer Science at DigitalCommons@Kennesaw State University. It has been accepted for inclusion in Master of Science in Computer Science Theses by an authorized administrator of DigitalCommons@Kennesaw State University. For more information, please contact digitalcommons@kennesaw.edu.

Implementation of Secure DNP3 Architecture of SCADA System for Smart Grids

A Thesis Presented to

The Faculty of the Computer Science Department

by

Uday Bhaskar Boyanapalli

In Partial Fulfillment

Of Requirements for the Degree

Master of Science, Computer Science

Kennesaw State University

July 2018

Implementation of Secure DNP3 Architecture of SCADA System for Smart Grids

Approved:

Dr. Donghyun Kim - Advisor

Dr. Dan Chia-Tien Lo– Department Chair

Dr. Jon Preston - Dean

In presenting this thesis as partial fulfillment of the requirements for an advanced degree from Kennesaw State University, I agree that the university library shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this thesis may be granted by the professor under whose direction it was written, or, in his absence, by the dean of the appropriate school when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of, this thesis which involves potential financial gain will not be allowed without written permission.

Uday Bhaskar Boyanapalli

Notice to Borrowers

Unpublished theses deposited in the Library of Kennesaw State University must be used only by the stipulations prescribed by the author in the preceding statement.

The author of this thesis is:

Uday Bhaskar Boyanapalli

1100 S Marietta PKWY,
Marietta, GA 30060

The director of this thesis is:

Dr. Donghyun Kim

1100 S Marietta PKWY,
Marietta, GA 30060

Users of this thesis not regularly enrolled as students at Kennesaw State University are required to attest acceptance of the preceding stipulations by signing below. Libraries borrowing this thesis for the use of their patrons are required to see that each user records here the information requested.

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Associate Professor Dr. Donghyun Kim of the College of Computing and Software Engineering College at Kennesaw State University. He consistently allowed this research to be of my work but guided me in the right direction whenever he thought I needed it. I would also like to thank the expert who involved in the validation survey for this research project: Dr. Junggab Son. Without his passionate participation and input, the validation survey could not be successfully conducted.

Finally, I must express my very profound gratitude to my parents, my brother and my cousins Avasarala Srinivasa Rao, MGV Padmavathi for providing me with motivation, inspiration and moral support throughout the years of study and research. This accomplishment would not have been possible without them. Thank you.

LIST OF FIGURES

Figure. 1: Smart Grid Architecture.....	12
Figure. 2: SCADA Architecture.....	14
Figure. 3: Programmable Logic Controller (PLC) usage in Industrial Control Systems	15
Figure. 4: Remote Terminal Unit (RTU) usage in Industrial Control Systems.....	16
Figure 5. Remote Terminal Unit (RTU) usage in Industrial Control Systems.....	18
Figure 6. Intelligent Electronic Devices Connection Topology.....	19
Figure 7. Intelligent Electronic Devices Connection Topology.....	20
Figure 8. DNP3 TCP/IP Connection with SCADA.....	21
Figure 9. DNP3 Connection Topologies.....	22
Figure 10. Distribution to Customer Substation Connection.....	30
Figure 11. Attack points on SCADA Architecture.....	31
Figure 12. Man in the Middle Attack (MITM).....	34
Figure 13. Distributed Denial of Service Attack (DDoS).....	35
Figure 14. Asymmetric Key Distribution for Users.....	36
Figure 15. Digitally Generated Certificate.....	37
Figure 16. Intermediate CA between users for verification.....	39
Figure 17. Securing SCADA Architecture using the Certificate Authority (CA) in DNP3...	40
Figure 18. DNP3 TCP/IP connection using Raspberry Pi on Wi-Fi Router.....	46
Figure 19. PKI Architecture of Distribution of Certificates.....	47
Figure 20. Verification of Outstation Certificate by Master as Intermediate CA.....	50
Figure 21. Verification of Outstation Certificate by Master as Intermediate CA.....	51
Figure 22. Verification of Outstation Certificate at Customer Substation by Master as Intermediate CA.....	52

LIST OF IMAGES

Image 1. Raspberry Pi 3 Model B Motherboard.....	45
Image 2. Certificate Authority Verification.....	53
Image 3. Outstation Certificate verification, and Private Key Password Request.....	53
Image 4. Master Private Key Password Request.....	54
Image 5. Master certificate decryption and connection with Outstation.....	54
Image 6. Outstation Certificate decryption and connection with Master.....	55
Image 7. Master receiving DNP3 messages.....	55
Image 8. Wireshark Capture.....	56
Image 9. TLS version 1.2 Encrypted Data Packet.....	56

ABSTRACT

With the recent advances in the power grid system connecting to the internet, data sharing, and networking enables space for hackers to maliciously attack them based on their vulnerabilities. Vital stations in the smart grid are the generation, transmission, distribution, and customer substations are connected and controlled remotely by the network. Every substation is controlled by a Supervisory Control and Data Acquisition (SCADA) system which communicates on DNP3 protocol on Internet/IP which has many security vulnerabilities. This research will focus on Distributed Network Protocol (DNP3) communication which is used in the smart grid to communicate between the controller devices. We present the DNP3 SAv5 and design a secure architecture with Public Key Infrastructure (PKI) on Asymmetric key encryption using a Certificate Authority (CA). The testbed provides a design architecture between customer and distribution substation and illustrates the verification of the public certificate. We have added a layer of security by giving a password to a private key file to avoid physical tampering of the devices at the customer substations. The simulation results show that the secure communication on the TLS layer provides confidentiality, integrity, and availability.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	5
LIST OF FIGURES.....	6
LIST OF IMAGES.....	7
ABSTRACT	8
1. INTRODUCTION.....	12
1.1 Smart Grid Introduction.....	12
2. SCADA SYSTEM BACKGROUND.....	14
2.1 Supervisory Control and Data Acquisition (SCADA).....	14
2.2 Programmable Logic Controllers (PLC).....	15
2.3 Remote Terminal Units (RTU).....	15
2.4 Intelligent Electronic Devices (IED).....	16
2.5 Human Machine Interface (HMI).....	17
2.6 Communication Protocols Used in SCADA – Power Grids.....	17
2.7 Modbus.....	18
2.8 IEC 61850.....	19
2.9 IEEE 1818 Distributed Network Protocol (DNP3).....	20
2.9.1 DNP3 Version.....	23
2.9.2 DNP3 IEEE Secure Authentication.....	23
2.9.3 DNP3 Secure Authentication versions and release.....	24
2.9.4 Version 5 adds cryptographic algorithms not supported by Version 2 as follows...	25
3 RELATED WORKS.....	26

4	RESEARCH METHODOLOGY.....	29
4.1	Model of SCADA Substation.....	30
4.2	Several Structure Scenarios.....	32
4.3	Attacks points and Vulnerabilities on DNP3.....	32
4.3.1	Malicious Insider.....	33
4.3.2	Tampering Physical System (Unauthorized User).....	33
4.3.3	Man in the Middle Attack.....	34
4.3.4	Distributed Denial of Service.....	35
4.4	Authentication Mechanism – Public Key Infrastructure (PKI).....	36
4.5	Location of CA in the Hierarchical model.....	38
4.6	Model Extension to other SCADA substations.....	39
4.7	Vulnerabilities Defended and Countermeasures.....	41
4.7.1	Tampering Physical Device.....	41
4.7.2	Man in the Middle (MITM) Attack.....	42
4.7.3	Malicious Insider.....	42
4.7.4	Distributed Denial of Service (DDoS).....	42
5	EXPERIMENTS	44
5.1	Test Environment.....	44
5.2	Configuration for RPI – DNP3.....	48
5.3	Implementation of PKI.....	48
5.4	Evaluation.....	52
6	CONCLUSION.....	57
7	REFERENCES.....	58

8	SOURCE CODE	63
---	-------------------	----

CHAPTER I

INTRODUCTION

1.1 Smart grid Introduction

Smart Grid referred to as the modernized power grids which connect different kinds of generations' plants to transmit and distribute to various customers connected in a grid-like architecture. Central aspects of this smart grid are to connect the smart technology which gives advanced control methods and secure system architecture for two ways of communicating with customers. The primary goal is to give information and control to every endpoint connected to the grid using internet on secure architecture. Smart grids use the latest technology to produce energy which reduces less carbon and uses new technology to optimize power for efficient energy resources. Smart Grid has 4 phases which are Generation, Transmission, Distribution, and Customers, which have their substation to control and manage the station using the Supervisory Control and Data Acquisition (SCADA) systems and appliances by A. Leonardi et al. [1]

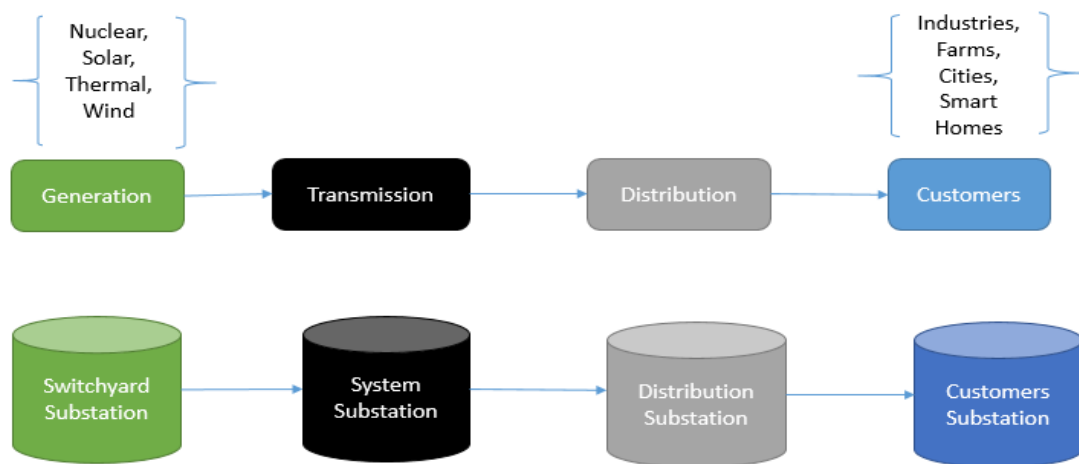


Figure 1. Smart Grid Architecture

Phase 1 of Generation of power is generated by Thermal, Solar, Nuclear and Wind, and at each generation plant, there is switchyard substation located to control, manage and send to the second phase. In the second phase, Transmission has system substation which receives electricity from the generation plant and sends to the third phase for Distribution. Here the distribution of electricity is distributed to the customer directly by remote control and calculation of smart technology with the help of internet. There are different types of customers which are of small scales such as rural cities, farms, small-scale industries, factories or businesses and others such as smart homes.

CHAPTER II

SCADA SYSTEM BACKGROUND

2.1 Supervisory Control and Data Acquisition (SCADA)

Power grid substation manages and automates using Supervisory Control and Data Acquisition (SCADA) which is a critical point in monitoring and controlling the process high-level management at the Industrial Control Systems. They include different kinds of controllers and peripheral devices to communicate on the Internet. They include different kinds of controllers and peripheral devices such Programmable Logic Controllers (PLC), Remote Terminal Unit (RTU), Intelligent Electronic Devices (IED), sensors, meters, embedded computers, Human Machine Interface (HMI) and field simulators.

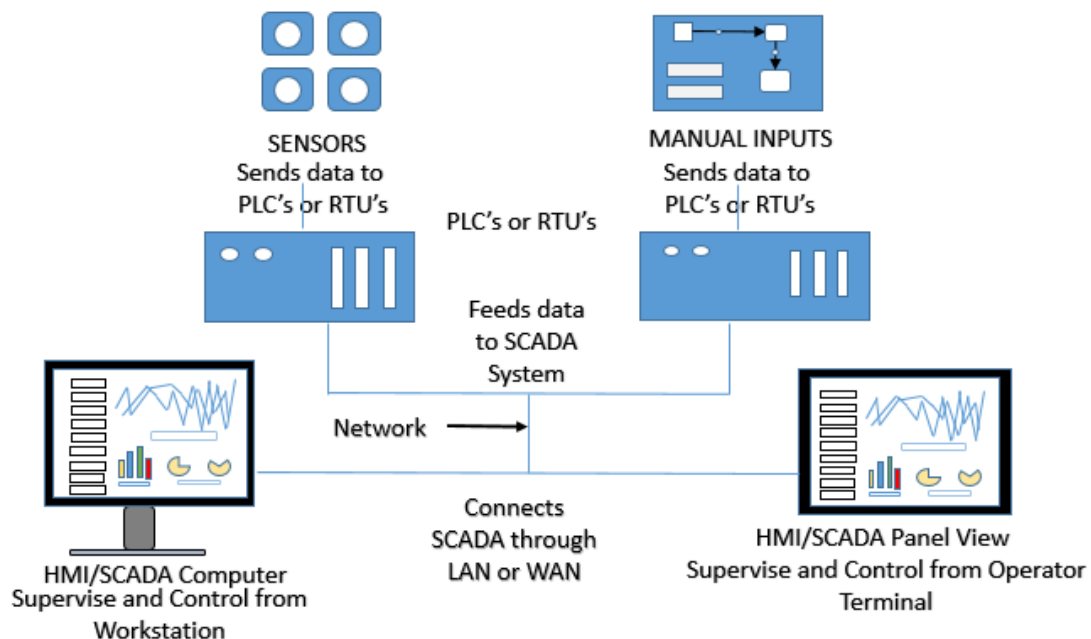


Figure 2. SCADA Architecture

2.2 Programmable Logic Controllers (PLC)

PLC is used in the industrial automation for the electromechanical process to control and automate the machinery connected to meters, sensors, alarms and collect the data from them based on pre-programmed parameters. This digital computer can be connected to a Human Machine Interface (HMI) and interact in real time to send and store the data. The physical connection can be communicated on several network devices and communication protocols such as IEC61850, Modbus and Distributed Network Protocol (DNP3) on Ethernet/IP, Ethernet TCP/IP, Modbus TCP/IP, and ProfiNet by Gordon Clarke et al. [22]. PLC is designed on vendor specific and can have a web server which can be connected to web browser also.

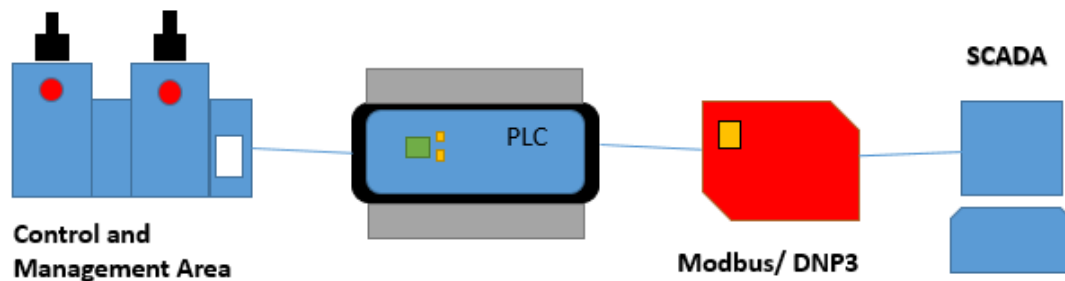


Figure 3. Programmable Logic controller (PLC) usage in Industrial Control Systems

2.3 Remote Terminal Units (RTU)

RTU is a field digital and analog controller which transmits messages to the master system using a SCADA system to control the connected objects. RTU's are used in different kinds of industries such as power, oil, gas, water, Hydro-graphic sewage systems and other

environment monitoring systems. RTU can communicate on multiple devices connected to them either in serial or using communication protocol such as IEC61850, Modbus and Distributed Network Protocol (DNP3) on Ethernet/IP to interface any software or SCADA systems. The typical architecture communicates as Master and slave/outstation.

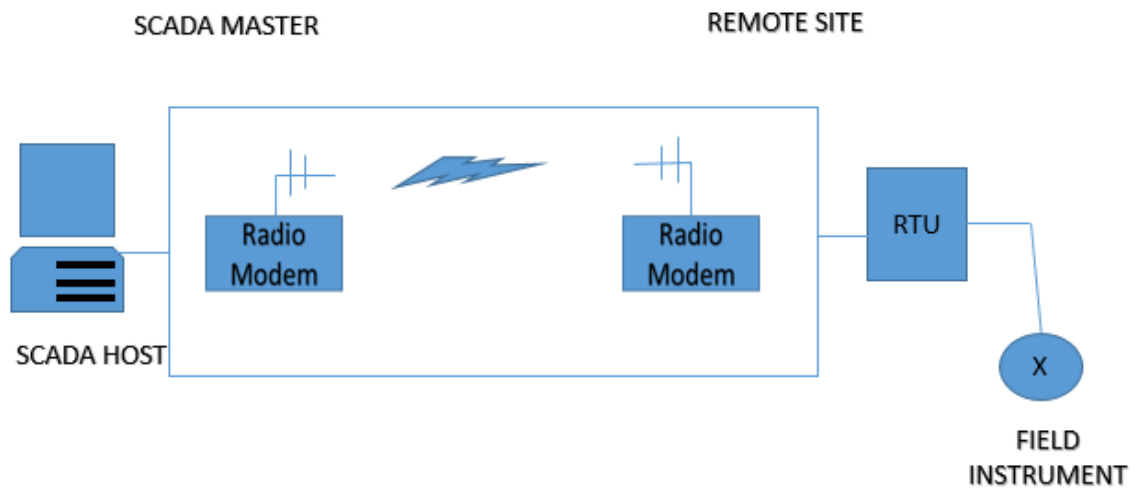


Figure 4. Remote Terminal Unit (RTU) usage in Industrial Control Systems

2.4 Intelligent Electronic Devices (IED)

IED's are connected to the meters, sensors, alarms to sense voltage, frequency and current to control them by raising voltage or decreasing to maintain in the desired level. RTU connected to the IED can control and send the command to control the field simulators. Digital Protective relay, such as microprocessors like IED's were used in the substation automation for advanced communication in the smart grid.

2.5 Human Machine Interface (HMI)

HMI is considered as an interface between a User and Machine to control and monitor by visualizing them in real time on graphical interface more specific to manufacturing or process control system. An HMI connected to PLC will be connected to the driver to monitor the field simulators to communicate and increase the efficiency by centralized control systems. HMI is used explicitly in SCADA systems to monitor and control data on the graphical user interface digitally. They can communicate with different kinds of protocols such as TCP/IP, DNP3, LAN, WAN, modem, and satellite.

2.6 Communication Protocols Used in SCADA – Power Grids

A communication protocol is a system of rule syntax that allows transmitting messages from one entity to another entity using a combination of hardware and software in semantics and synchronization. These communication protocols are authorized as a standard for electronic smart grids by the National Institute of Standard and Technology (NIST) and IEEE recommended [28]. These protocols can communicate on different interfaces such as Transport Control Protocol (TCP)/Internet Protocol (IP), Ethernet, Bluetooth, 3G which was presented in Byron Flynn [2]. The Standard communication protocols which smart grids use in the communication of peripheral devices are IEC 61850, Modbus, and IEEE 1815 (DNP3).

2.7 Modbus

Modbus is an Ethernet, Internet protocol suite used as a communication protocol for industrial applications used for PLC and RTU in the SCADA systems. It is updated and managed by the Modbus organization when the Schneider Electric gave their rights to that organization. They are used for the serial connections interfaces such as RS 232 and RS 422 and RS 485. They communicate with different protocols such as TCP/IP, UDP, ASCII, Modbus RTU and some vendor-specific protocols designed for the clients.

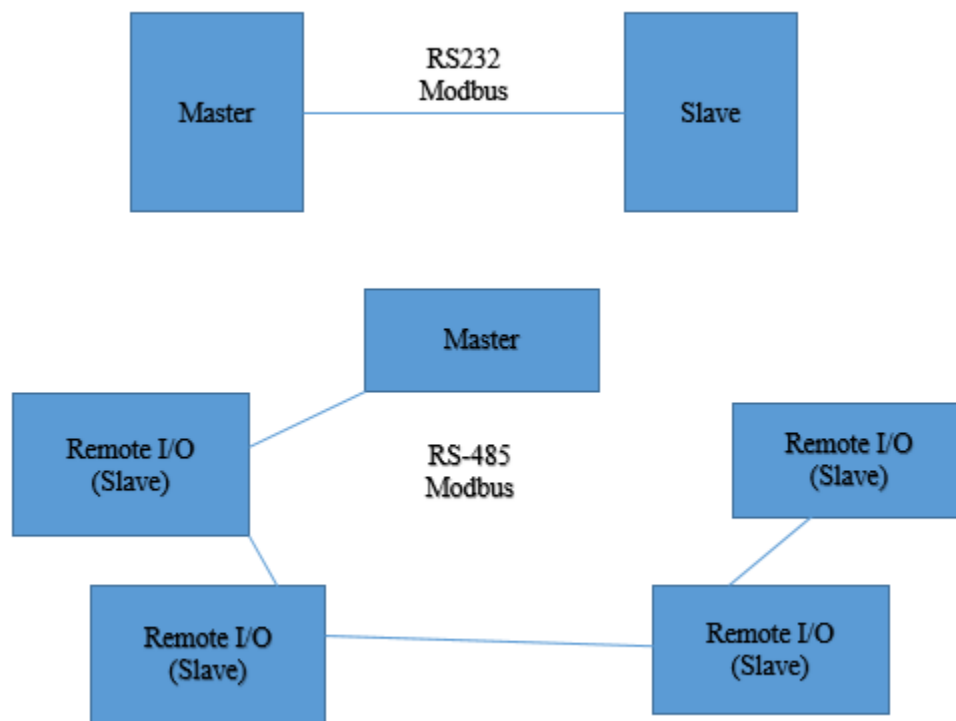


Figure 5. Remote Terminal Unit (RTU) usage in Industrial Control Systems

2.8 IEC 61850

This protocol can be mapped to other protocols to be used for the intelligent electronic devices used at the electrical substations and is a part of the International Electrotechnical Commission (IEC) technical committee. IEC 61850 run on LAN, TCP/IP, Ethernet and other high-speed networking to transfer data to be mapped and designed as global standard communication for different countries in electric utilities.

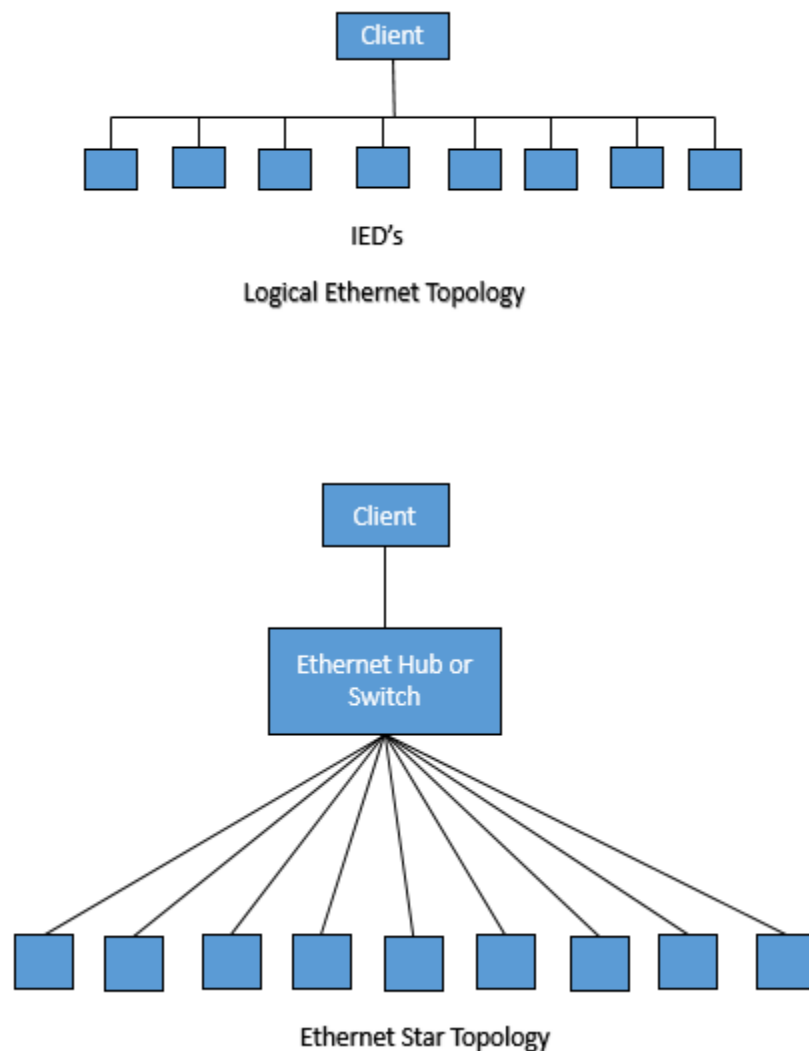


Figure 6. Intelligent Electronic Devices Connection Topology

2.9 IEEE 1818 Distributed Network Protocol (DNP3)

DNP3 is an open standard protocol used as a communication protocol between process automation and control system in electric and water substations. Designed by the International Electrotechnical Commission (IEC) for the OSI layer three used in the field simulators as a communication protocol. The multilingual feature of DNP3 that allow any proprietary protocol to communicate with it allowed many to upgrade their network to DNP3. It can work on different communication networks such as LAN, WAN, Ethernet, TCP/IP is a crucial feature to send messages to its geographical locations. It plays a crucial role in SCADA system communication which is used by SCADA master, RTU's, IED's, PLC's and other inter-master station communication.

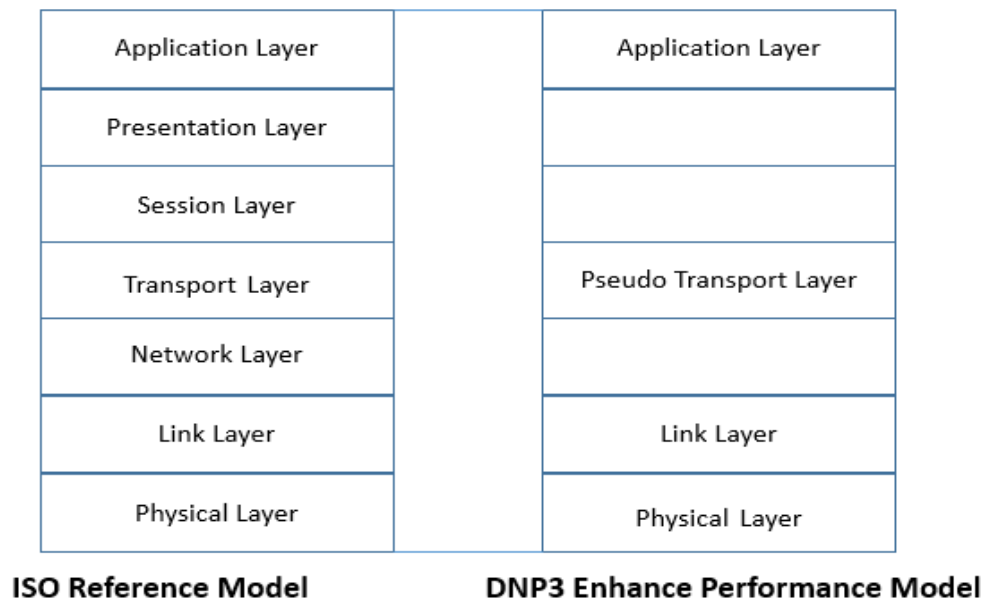


Figure 7. Intelligent Electronic Devices Connection Topology

The model is based on the Open Systems Interconnection (OSI) model that is a seven-layer model and contains an application layer, data link layer, and physical layer. DNP3 also uses these three layers, plus an additional layer called the pseudo-transport layer which performs the limited functions of the transport layer and network layer of the OSI model.

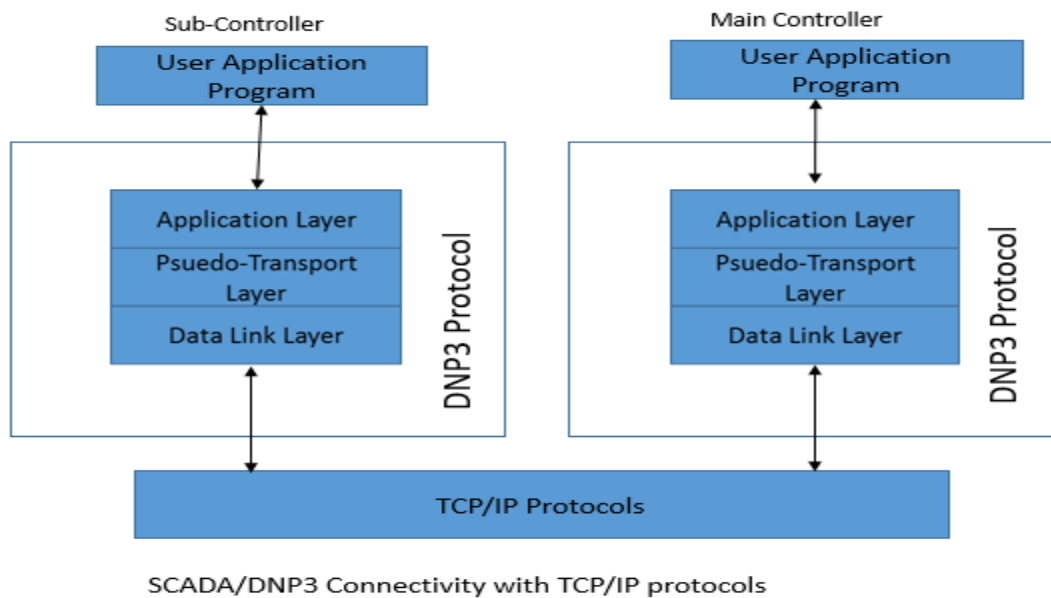


Figure 8. DNP3 TCP/IP Connection with SCADA

DNP3 can be designed in a few typical architectures such as one-on-one, Multi-drop, Hierarchal, Data Concentrator. At the top is a simple one-on-one system having one master station and one outstation. The physical connection between the two is typically a dedicated or dial-up telephone line. Multi-drop design. One master station communicates with multiple outstation devices. Conversations are typically between the master and one outstation at a time. Multi-drop communications are peer to peer. In the Hierarchical

architecture, the device in the middle is often termed as sub-master in DNP3 protocol primer [27].

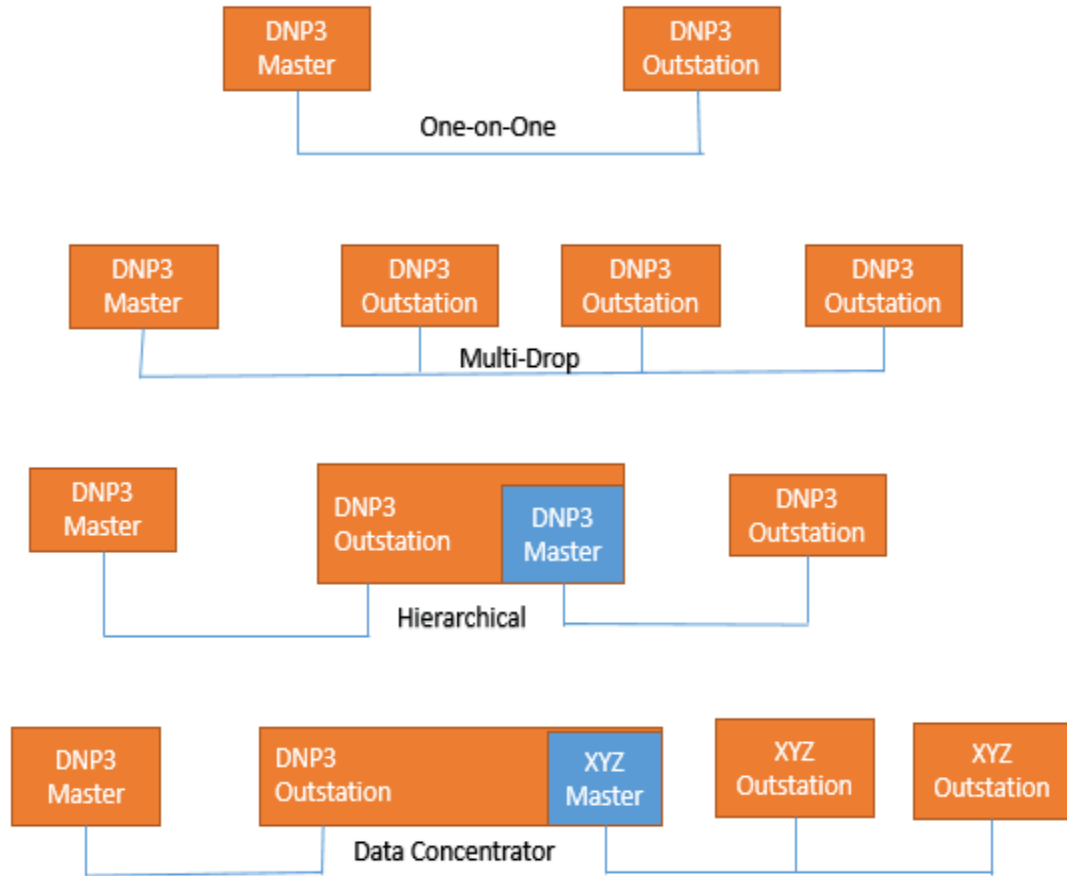


Figure 9. DNP3 Connection Topologies

DNP3 supports physical communications such as RS 232/485 and also TCP/IP. It supports sending data as blocks and file transferring can be done in large data blocks. DNP3 is an open standard which enables the organization to design and secure the communication according to the industry needs. Evaluation of the performance and cyber application in smart grid were analyzed in the research proposed by Alcides Ortega et al. [17].

2.9.1 DNP3 Version

1. 1379-2000 - IEEE Recommended Practice for Data Communications Between Remote Terminal Units and Intelligent Electronic Devices in a Substation.
2. Distributed Network Protocol (DNP3) - IEEE Standard for Electric Power Systems Communications.
3. Distributed Network Protocol (DNP3) - 1815-2012 - IEEE Standard for Electric Power Systems Communications.

2.9.2 DNP3 IEEE Secure Authentication

DNP3 security authentication (SA) identifies the message source and assigns application layer functions and data objects that allow the device to authenticate DNP3 communication messages and verifies that the message has been transmitted without modification. This protocol is designed to be highly reliable but is not designed to be secure from hackers and other malicious attacks that may interfere with the control system to avoid using critical infrastructure. The protocol was designed to be very stable but not designed. It is safe from hackers and other malicious attacks that can interfere with the control system to prevent the use of critical infrastructure. DNP3 Secure Authentication from Jacques Benoit [14] provides services to authenticate the sender and the content of messages. DNP3 Secure Authentication is based on the IEC 62351-5 security standard.

The “de facto reference implementation of IEEE 1815-2012” OpenDNP3 is the only Open Source DNP3 implementation, and it is maintained by Adam Crain [9] of Automatak, who has been adding TLS layer and DNP3 SAV5 support to his stack.

2.9.3 DNP3 Secure Authentication versions and release

DNP3 Secure Authentication Version 2 (SAv2) was first released as part of IEEE STD 1815-2010. DNP3-SAv2 has been deprecated and superseded by DNP3 SA Version 5 (SAv5) included in the most recent release of the DNP3 standards (IEEE Std. 1815-2012) [31].

DNP3-SA is based on the international standard IEC 62351-5 which was test by Raphael Amoah et al. [5]. Which in turn is based on several different standards issued by the International Standards Organization (ISO), International Electrotechnical Commission (IEC) the Internet Engineering Taskforce (IETF) and the U.S. National Institute of Standards and Technology (NIST) and Richard Duncan from SANS Institute [13]. Secure Authentication Version 5:

2.9.4 Version 5 adds cryptographic algorithms not supported by Version 2 as follows

1. Instead of minimum length 4 of an HMAC can change to 8 octets in DNP3
2. It makes SHA-256 a mandatory hash algorithm, and the default.
3. It makes it a requirement that configuration can disable the use of SHA-1.
4. It changes the mandatory TLS cipher suite to one supported by TLS version 1.2.
5. It clarifies which pseudo-random number algorithm should be used.
6. It permits the use of the AES- 256 Key Wrap algorithm optionally.

7. It optionally permits the use of the AES MAC algorithm for calculating MACs. Using this algorithm places some additional rules on the rest of the protocol by IEEE [31].
8. The Version 5 specification now uses the generic term MAC instead of HMAC since HMAC is a specific type of MAC which was mentioned by Cas Cremers et al. [4] and by IEEE standard [30]:

In this research, Distributed Network Protocol 3 (DNP3) is focused as the central aspect of the communication in SCADA systems because it is considered as the most commonly used power grid protocol in North American Utilities for distribution and DER communications. Modbus protocols used in the power grids are vendor specific and are commercially licensed whereas IEC 61850 is used less when compared to DNP3 as it is an open and public protocol which can be designed according to the smart grid requirements. DNP3 Secure Authentication version 5 is focused on this research as it is the new standard and supports Public Key Infrastructure (PKI). This model is a novel work where PKI implementation on DNP3 SAV5 and design of the architecture to how securely identify the authenticity of machines are communicating.

CHAPTER III

RELATED WORK

This chapter mainly focuses on the research done on the communication protocol used in the power grid, DNP3 and its vulnerabilities which are penetrated by the attackers. Recommendations for securing the architecture from research done on DNP3 are a symmetric connection and TCP/IP communication which are related to the research presented in this thesis.

In [1], A. Leonardi, K.Mathioudakis, A. Wiesmaier, and F. Zeigar has described the smart grid architecture and technology and how the substation automation plays a significant aspect for the security vulnerability in the power grid mainly because of the remotely accessible devices placed within substations and customer locations on outside network infrastructure. Specifically, this paper discussed the communication protocol IEC 61850 and utility applications that typically run at the control center in transmission or distribution center which lacks the physical and cybersecurity.

In [2] Byron Flynn, mentioned different kinds of real-world architectures used in SCADA which include various methods of user authentication and secure access to the substation connections and servers. He mentioned their advantages and disadvantages on how security would be a vital part in secure system operation and centralized access right control.

In [3] Ihab Darwish, Obinna Igbe, Orhan Celebi, Tarek Saadawi, Joseph Soryal have highlighted different vulnerabilities and security threats in DNP3 as real-time automation. They have shown TCP connection used in DNP3 which was released in 2010 and experimented on virtual computer environment to show its security issues and gave suggestion to use Intrusion Detection System which will be useful to identify the cybercriminal targeting the substations in the Smart Grid Infrastructure.

In [4] Cas Cremers, Martin Dehnel-Wild, Kevin Milner, have considered DNP3-SAv5 and symbolic modeling on the protocol's three sub-protocols. The cross-protocol attacks which are mentioned by them in their previous research have been analyzed again and were unsuccessful on the SAv5 version due to its security properties. This research has to lead to different recommendations for improving the future versions of DNP3 in building block for power substation communication.

In [5] Raphael Amoah, Seyit Camtepe, Ernst Foo, have proposed the lightweight security scheme for broadcasting mode communication based on hash chain using update key process on DNP3 SA. The proposed scheme is verified using the most common protocol attacks such as modification, replay and injection attacks using the symmetric key as DNP3-SA key update process.

In [6] Carlos Lopez, Arman Sargolzaei, Hugo Santana, Carlos Huerta, have researched smart grid infrastructure for power and generation for the integration of advanced

communications and networking for monitoring and controlling them. They have revised different kinds of cyber threats associated with these critical infrastructure SCADA systems and these attacks are identical in attack but applied according to their different types of smart grid. A different number of computational, and experimental algorithms are proposed which can improve the security in the smart grid.

In [7] Dongsoo Lee, HakJu Kim, Kwangjo Kim, Paul D. Yoo, have simulated the DNP3 SA with pre-shared keys using either symmetric or asymmetric key and find various ways to attack this system. Lack of confidentiality made them propose the DNP3 Authenticated Encryption for both authentication and encryption. Their testbed was tested on three devices and also Prince Algorithm as not implemented to test and to get more precise and helpful results.

In [8] Samuel East, Jonathan Butts, Mauricio Papa, and Sujeet Sheno have done their research on the DNP3 and their vulnerabilities which provided the attack taxonomy which clarifies the nature and scope of the threats to DNP3 systems. The attack taxonomy also provided insights into the relative costs and benefits of implementing mitigation strategies.

In [11] Marcio Andrey Teixeira, Tara Salman, Maede Zolanvari, Raj Jain, Nader Meskin and Mohammed Samaka have done research on network packets which are cyber-attacks done on a test bed environment and this dataset is trained with Machine Learning algorithms to detect the attacks in the real time.

CHAPTER IV

RESEARCH METHODOLOGY

In this chapter, we demonstrate the research methodology and model of the proposed secure architecture of DNP3 SAv5 using Public Key Infrastructure for authenticity verification using Public Certificate or Public Key of entity it is communicating. Also, illustrate the real world system with their vulnerabilities and simulated attacks on DNP3 Secure Authentication and analyze these vulnerabilities. Further, explain the proposed scheme Public Key Infrastructure (PKI) which was introduced by Prof. More V.N. [12] and enhancement of DNP3 security using the public certificate verification using the root certificate authority (CA) or a substitute CA.

In the smart grid, we focused on a particular aspect where the distribution substation connected to customer substation where at this state, the design architecture of the connection varies with the customer substation with the size of the customers. The customer substations can be of different types such as Industrial plants, smart homes, cities or others. Substation model can be of different types depending on the size of the city or a smart home community.

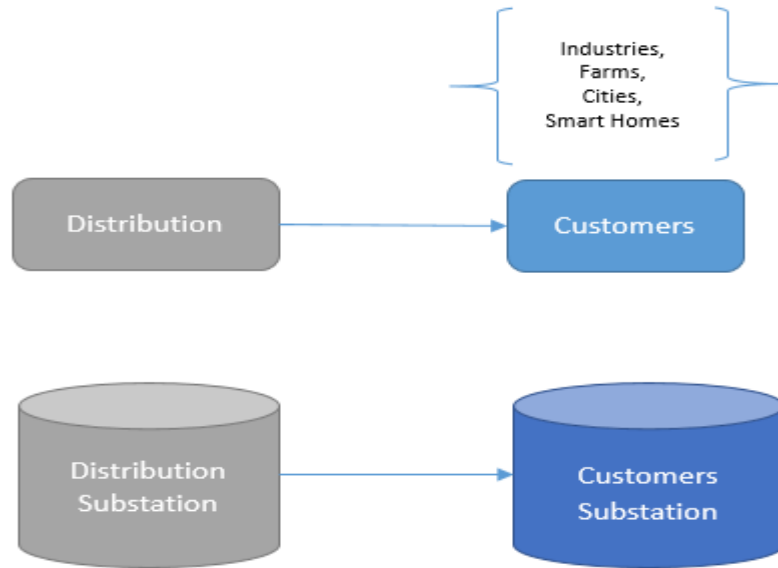


Figure 10. Distribution to Customer Substation Connection

4.1 Model of SCADA Substation

The Hierarchical model used in the SCADA substation where the master connected to a set of outstations and some of the outstations can also act as a Master in this model which was shown by Abdulmalik Humayed et al.[10]. In the distribution substation model, it has a SCADA master controlling and collecting the data in the data storage with remote connections connecting on Local Area Network (LAN). The below hierarchical model depicts the distribution substation connecting to a smart home connected to the smart grid. The connection between the smart homes and distribution substation as illustrated below, and the protocol used to communicate here is DNP3 on internet/IP address. DNP3 communication on TCP communication has the side effect of cybersecurity issues which enables a hacker to penetrate the weak points at the SCADA system which are surveyed in the Wenye Wang et al. [21].

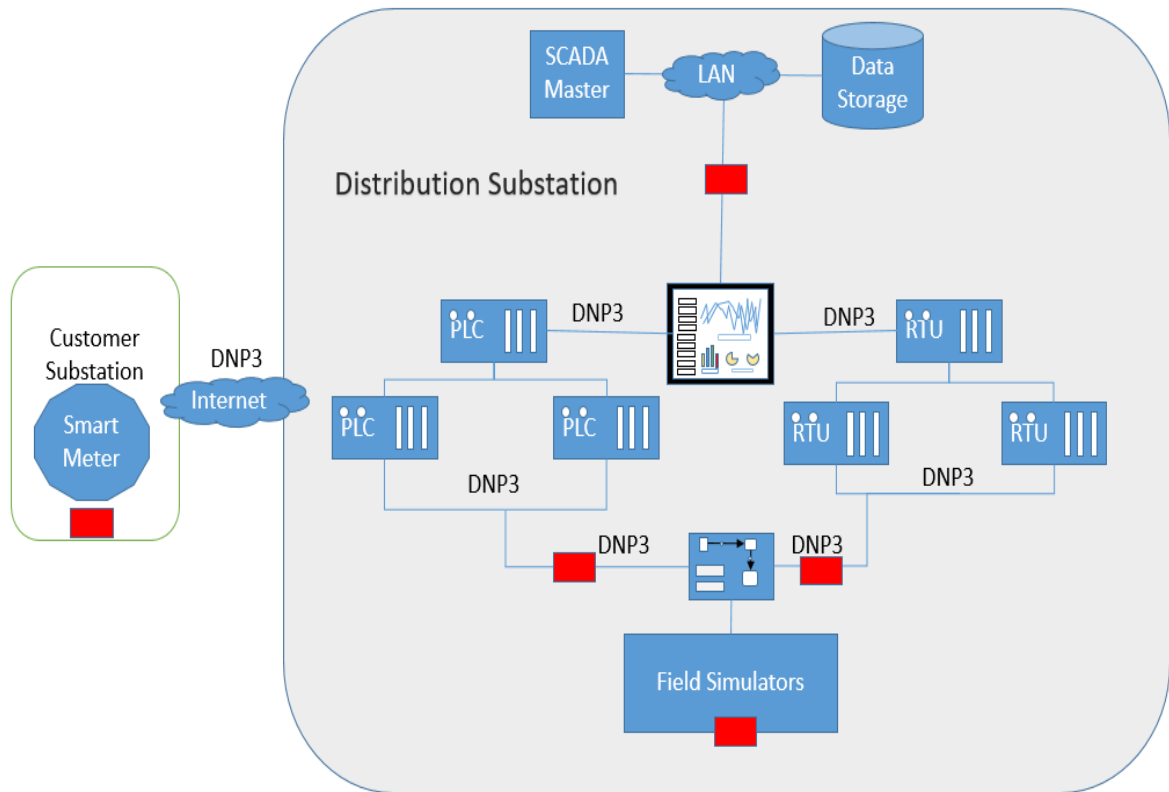


Figure 11. Attack points on SCADA Architecture

The attack point shows the vulnerable places where hackers penetrate the system. Connections on LAN area connection on DNP3 using TCP/IP communication would allow a hacker to make a replay or MITM attack to spoof the messages communicating on DNP3 which are introduced in Chih-Che Sun et al. [15]. The customer substation should also be secured from the physical attacks which can have a connection to the SCADA system which allows them to penetrate them from remote location remotely.

4.2 PKI Structure Scenarios

In SCADA there were many types of connections, and our design model for DNP3 SAv5 with Public Key Infrastructure (PKI) can implement for all using the Certificate Authority (CA) as the third party or as can be used in the SCADA master. In this model, Public certificate plays a significant part in the verification of the entity during a physical or a malicious attack using the internet on the device communicating on DNP3. The root Certificate Authority (CA), the interim or Sub-CA can verify the entity sent by Master or Outstation it is communicating. By this verification, the entity can know the authenticity of the device and the certificate here is password protected. PKI also provides the TLS encryption, authentication, and authorization which made our model secure and verification of the certificate or public key can validate the authenticity of the entity.

4.3 Attacks points and Vulnerabilities on DNP3

The SCADA systems communicating on DNP3 have physical and cyber system challenges which are addressed by Samuel East et al. [8] such as passive network reconnaissance, baseline response reply, and rouge interpreter. These frequent attacks can modify or fabricate the DNP3 messages, and the entity assumes as these messages are valid. They have introduced the three common attacks with 21 attack instances which can potentially operate malicious operations on the entity devices. Irfan et al. [1] studied the attack instances and discussed the attacks mitigations for the configuration cipher attacks, length outer flow attack, and flag function unavailable attack and also Man in the middle attacks. Humayed et al. [3] have discussed the operating system vulnerability in PLC and RTU at

remote stations, communication vulnerabilities such as 23 attacks which exploit the absence of encryption, authentication, and authorization. Yilin Mo [4] have explained different attack entities where an attacker can penetrate using such as the malicious media, network-based intrusion, compromised supply chain, and malicious insider. Confidentiality of meter usage and commands, the integrity of price information, penetrating the device to perform a Denial of Service. The attack trees are also studied in various research and introduced in Eric J. Byres et al. [16]

4.3.1 Malicious Insider

A legitimate user, employee or an ex-employee who has access to the physical devices, SCADA systems, or has the VPN access to the field devices and uses for his malicious purposes can call as a malicious insider. The malicious insider is dangerous and also a threat to the smart grid which was shown in by Adnan Anwar et al. [20]. The access control should also monitor who can access the network and authenticity of the person using needed to be verified.

4.3.2 Tampering Physical System (Unauthorized User)

Any unauthorized user trying to penetrate the device physically would be a threat to the whole SCADA system and the smart grid. The devices which are at the customer substations such as smart meters or remote RTU's, PLC or smart homes are vulnerable to physical tampering. A remote device when penetrated can be used as a malicious

entity to destroy or can alter the data which is communicating to other substations. The mitigation to secure these remote devices also plays a significant role in securing the system

4.3.3 *Man in the Middle Attack*

Darwish et al. [3] have an experimental study on the DNP3 performing penetrating testing such as Man in the Middle (MITM) attack by simulating the attack on an Ethernet switch. This vulnerability would allow an attacker to intercept and inject a packet for malicious purposes. Eavesdropping is an example of a MITM attack where an attacker relays messages between connections and make them believe that they are communicating with each other by Rajendra Kumar Pandey et. Al. [19]. Mitigation of MITM would be mutual authentication and Transport Layer Security (TLS) with encryption as well as issuing digital certificates. When two entities are talking to each other, and a malicious thing intercepts and eavesdrop the conversation in the form of sniff and spoof for malicious purposes. Digital Certificates can prevent this.

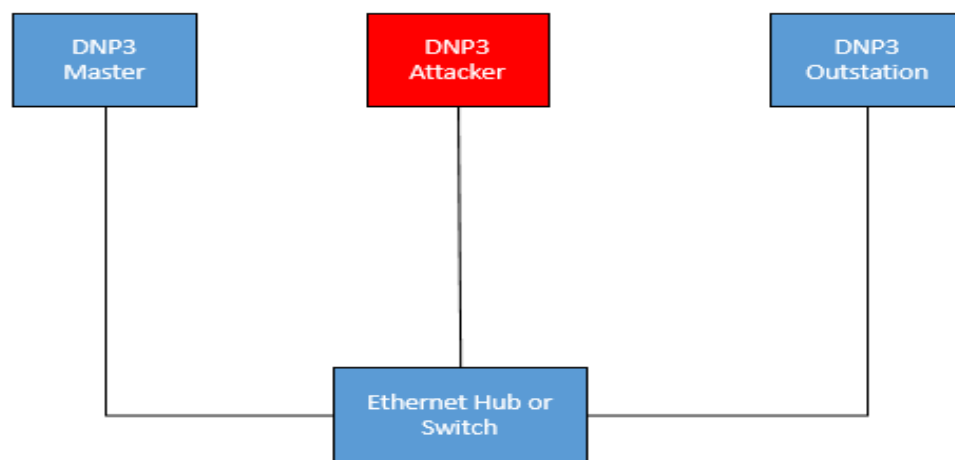


Figure 12. Man in the Middle Attack

4.3.4 Distributed Denial of Service

East et al. [8] have warned about the Distributed Denial of Service (DDoS) as a cyber-attack which the attacker can make the machine unavailable for intended users temporarily. DDoS is typical can be accomplished by flooding messages on the targeted device using the network and which make the application for denial of service. The simplest type of botnet attack is to flood an ordinary Web site with bogus messages, blocking or slowing the normal flow of information. These “denial of service” attacks could also be used to slow traffic moving between the control station and substations.

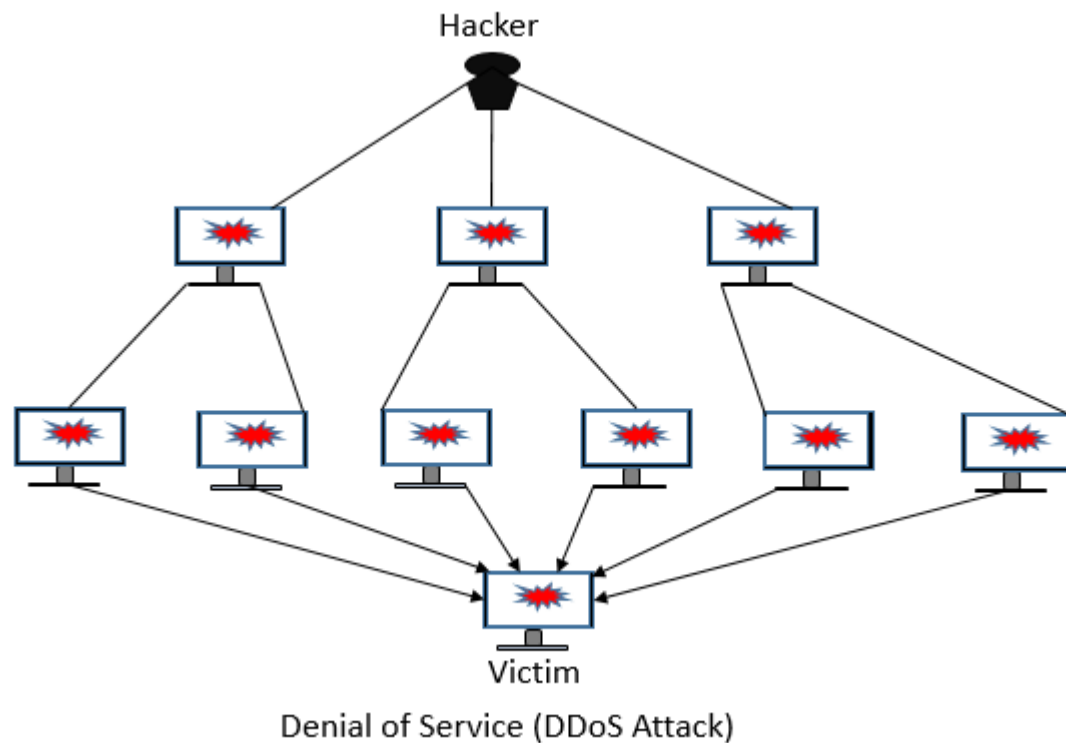


Figure 13. Distributed Denial of Service Attack (DDoS)

4.4 Authentication Mechanism – Public Key Infrastructure (PKI)

DNP3 SAV5 supports Secure Authentication which allows the end to end cryptogaic authentication on the TLS layer with advanced encryption. In our design we have used Public Key Infrastructure (PKI) which provides the SCADA users to produce, manage, distribute, store, revoke digital certificates using public certificates. PKI is used in many platforms for secure electronic transformation for a range of activities such as e-commerce, internet banking and also in the confidential email. PKI relies on asymmetric key encryption using public key cryptography.

In our design, we have used public key cryptography with asymmetric key encryption. Each user communicating on the asymmetric key encryption will choose their private key, and later a public key is obtained from the private key cryptographically. One key in the pair is shared between every entity, and it is called as a public key, and the private key is kept secret. One key is used to encrypt, and another key is used to decrypt and many protocols such as digital signature, SSH relies on asymmetric cryptography.



Figure 14. Asymmetric Key Distribution for Users

PKI issues digital signatures which refers to a set of algorithms and encryption methods which used to determine the authenticity of the document, software or message. A digital signature provides non-repudiation in which a user cannot deny sending the message and integrity of the message was not altered in the transit. There are two main properties which make PKI secure. First, the authenticity of signature from the message and fixed private key is verified using a public key, and secondly, it is computationally infeasible to generate a valid signature for an entity without the entity's private key. The author of the message to attach a code that acts as a signature is the authentication mechanism in the digital signature. The digital signature has various details of the entity which is created by the certificate authority (CA) which is the root of handling the PKI infrastructure.

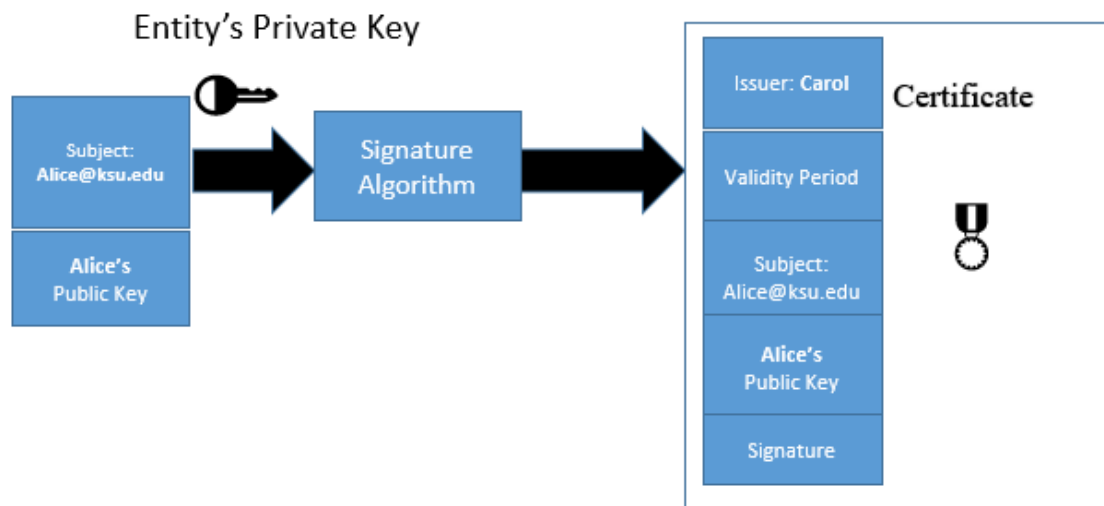


Figure 15. Digitally Generated Certificate

Certificate Authority (CA) is a trusted entity which can be third party or a trusted entity such as a SCADA master in the design that issues electronic documents to the entity that verify a digital identity on the internet. The electronic certificates typically include validity,

time, owner's name, and other information of the public key owner. Usually, operating systems (OS) and Web browsers maintain the list of trusted CA root certificates to verify that the CA issues and signed the certificates.

4.5 Location of CA in the Hierarchical model

In the Hierarchical model, the smart grid manager would be acting as the root CA which trust anchor of the whole secure communication and digitally signed and verifying the certificates. The CA's infrastructure within the auditable requirements is a complex task which includes operational elements, hardware, software, policy framework and practice statements, auditing and security infrastructure and personal. The CA's should not issue the digital certificates directly from the root distributed but instead by having the interim or sub-CA's in the architecture. Interim or Sub CA can be a SCADA master, and other interim CA can be placed in the entity which is communicating to the remote substations.

The potential exposure of Root CA can be a weakness which is vulnerable to attackers, and it should be out of reach of the internet. The intermediate CA in the SCADA master will verify the certificates which the PLC or RTU connecting. The remote RTU or a smart meter at the customer substations has the public certificate and the private key secured in the meter.

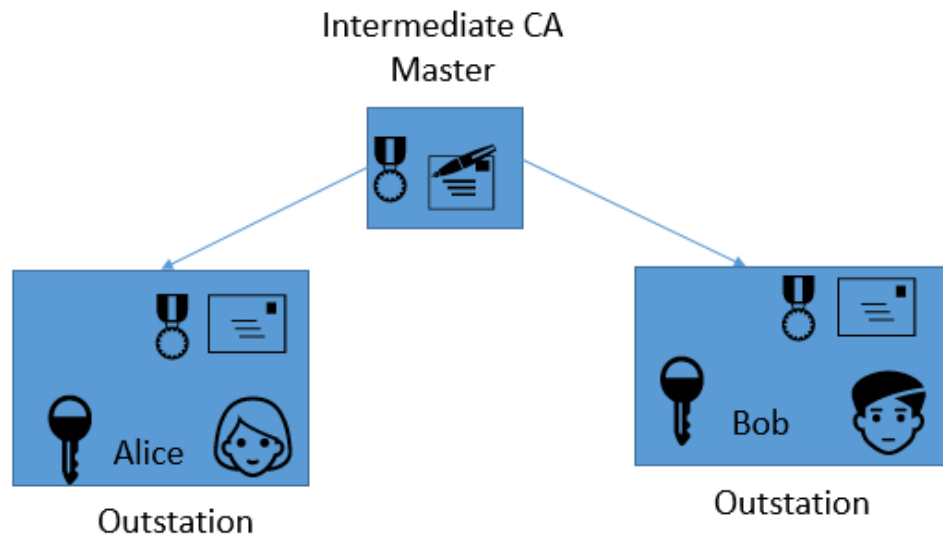


Figure 16. Intermediate CA between users for verification

4.6 Model Extension to other SCADA substations

The architecture tested was on hierarchical, and one to one model where the DNP3 SAv5 worked was on IP address on Wireless Area Network (WAN). The communication we worked was Master, and Outstation like architecture and the packets were sent on Transport Layer Secure (TLS). This is a standard model which is used in the SCADA architecture from Keith et al. [1] in which we added a DNP3 SAv5 using PKI with certificate verification. The authorization of the systems to connect and communication is managed by the Intermediate Certificate Authority (CA). The communication of the system is done TLS layer which provides Confidentiality, Integrity, and Availability of the device. This also authorizes the authenticity of the device, and an added a layer of security to the private key to avoiding physical tampering of the smart meter at the customer substation.

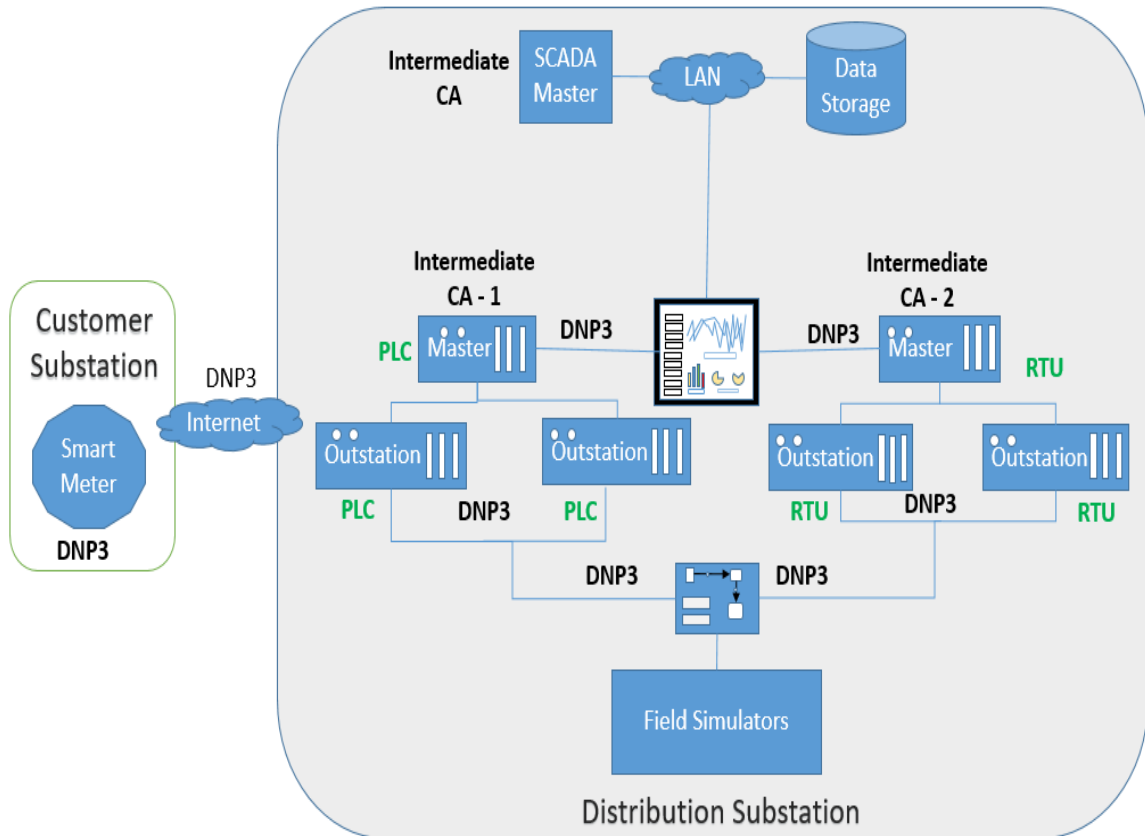


Figure 17. Securing SCADA Architecture using the Certificate Authority (CA) in DNP3 Protocol

In [24], James H. Graham et al. proposed some cost effective implementation alternatives with encryption and SSL/TLS layer report evaluation and in this research, we have implemented the TLS 1.2 using the PKI and certificate verification.

4.7 Vulnerabilities Defended and Countermeasures

Vulnerabilities in the DNP3 can be mitigated by the securely designing the architecture using the SAV5 with Public Key Infrastructure (PKI). The communication vulnerabilities of 23 attack instances can be defended by the design architecture proposed. The possible DDoS, MITM, and tampering of the devices comprises of the 23 attacks and will discuss the how the model can defend these attacks.

4.7.1 Tampering Physical Device

Tampering of the device physically can be mitigated with the securing the physical devices with a password and access control to the device should be assigned. The password protected device if cracked and have access to the private key and public certificate there is a possible chance of penetrating the system and an Event Buffer flooding attack by Dong Jin et al. [23]. . So in our design, we have given the password for the private key that if any device to connect with their public certificate, the private key needs a password to be operated. CA is needed to audit successful logins and login attempts and verification of certificates. If the physical device is tampered and try to authenticate with the device the public certificate should be verified with the Certificate Authority (CA).

4.7.2 Man in the Middle (MITM) Attack

If there is a Man in the Middle and uses a malicious certificate, the CA will ignore and block the Man in the Middle (MITM). The certificate presented by the attacker will be tested by the intermediate CA in the SCADA master or the PLC's and will notify the device if the CA signed the certificate or not. Verification of the certificate will allow the design to have an added advantage of security and will block the Man in the Middle. The communication is in TLS secured and encrypted which won't allow an attacker to sniff or spoof the packet.

4.7.3 Malicious Insider

Malicious entity acts as a threat to the SCADA system, such as an employee, former employee, business associates who have access to information regarding the organization's security practices. The threat can be malicious software inside the organization SCADA system which is also called as logic bombs. Access control in the SCADA systems and authorization of the client should be managed to mitigate the malicious insider. If an ex-employee has access to the SCADA systems, he can penetrate the system for malicious purposes.

4.7.4 Distributed Denial of Service (DDoS)

Establishing a response plan and blocking the IP addresses at the network level to protect these attacks. Recommendation for these attacks would be an Intrusion

Detection system with a set of rules and access control of the IOT devices. Distribution of intermediate CA to the other devices will avoid the single point bottleneck and stops the DDoS attack from concentrating on the target.

CHAPTER V

EXPERIMENTS

In this chapter, we demonstrate the research methodology and design of the Secure Architecture version 5 of DNP3 using Public Key Infrastructure for authenticity verification using Public Certificate. Firstly, the testbed will be explained and how the installation of DNP3 open protocol from GitHub and the TCP communication. Secondly, configuration changes according to the testbed which is used will be explained, and in the third, the implementation of PKI - the creation of keys and certificate will be explained using the OpenSSL. In the fourth, the implementation of verifying the entity using a public certificate will be explained. Lastly, we will show the design and results of the implementation of PKI. The evaluation of the design will show the screenshots of every step of outputs in the execution phase.

5.1 Test Environment

We have made our test environment using a UNIX based operating system - Noobs, running on Raspberry PI 3 model B with Wireless Area Network (WAN). The hierarchical model consists of multiple master/outstation like architecture in SCADA systems, and Raspberry PI model can run as an embedded system. The master outstation motherboard was Raspberry PI 3 model B with the configuration of 802.11n Wireless LAN, 10/100Mbps LAN Speed, Bluetooth 4.1, 4 USB ports, 40 GPIO pins.

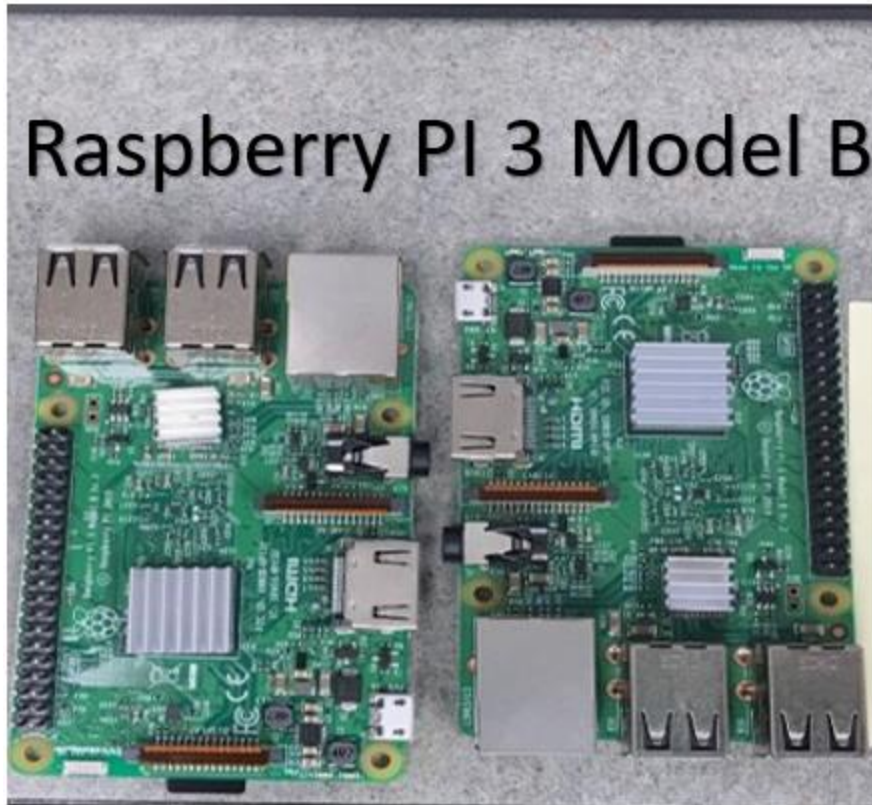


Image 1. Raspberry Pi 3 Model B Motherboard

We have opted for the Opendnp3 [33] which is a reference to DNP3 (IEEE – 1815) protocol and is an open source from the GitHub [33]. Many DNP3 protocols are for commercial use, and Automatak with the help of the author J. Adam Crain et al. [9] made this protocol as open source. We have used the C++ version of DNP3 with GCC and G++ compilers [34] and to build the system we have used the CMAKE [35]. ASIO is used as cross-platform for input and output communication for the Linux build environment. Firstly, the operations were tested for the TCP communication in the below architecture where the communication is open and with fewer security features.

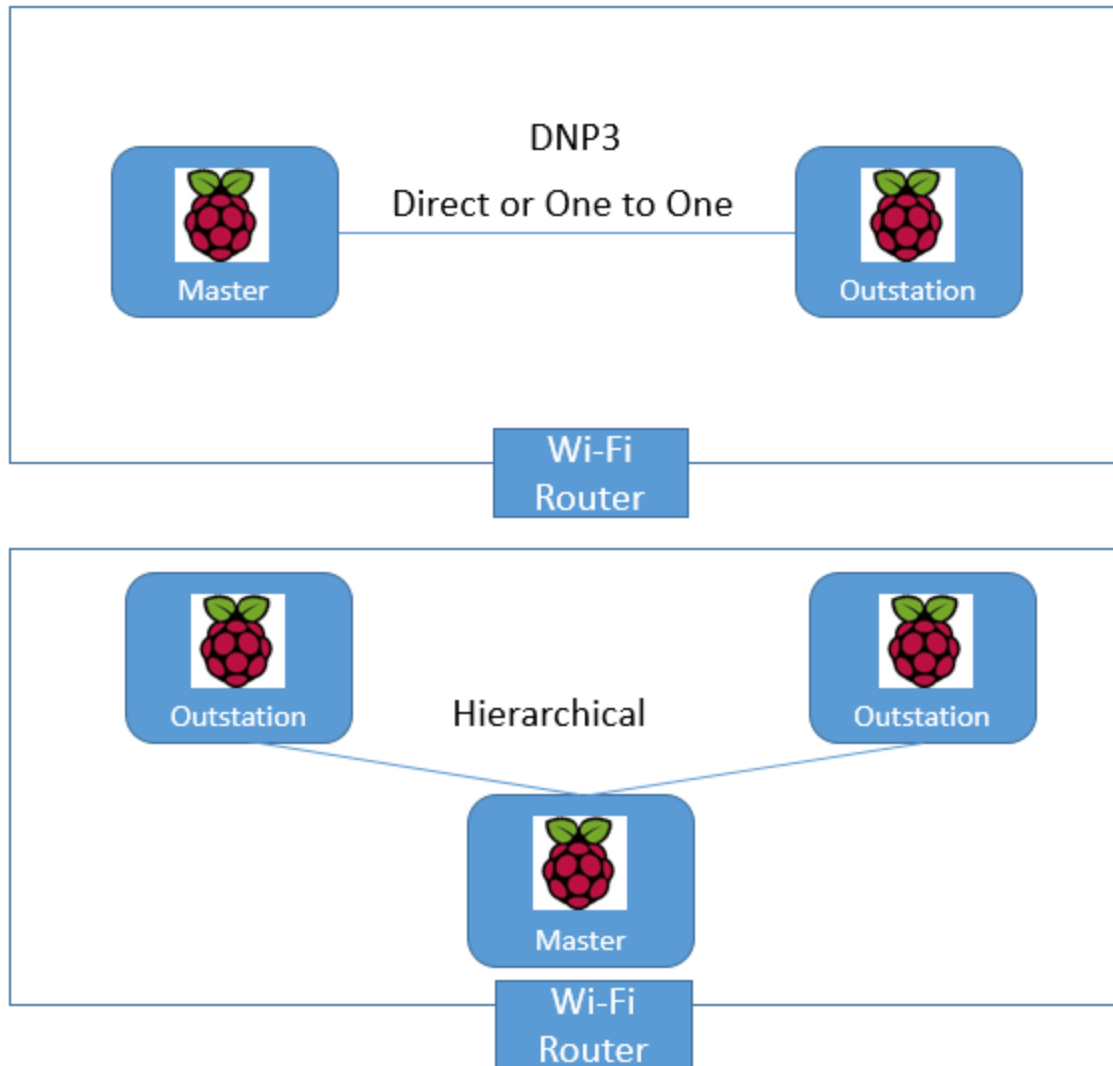


Figure 18. DNP3 TCP/IP connection using Raspberry Pi on Wi-Fi Router

The OpenSSL cryptography library [36] is used for the Key generation, digital certificates and TLS communication over the computer networks such as dynamic key generation for securing the data with encryption which was propose by Neetesh et. al [25]. It is widely used in the internet web servers, credit card transactions and communication on network securely with the Public Key Infrastructure (PKI). PKI is used to generate, distribute, store

and revoke the digital certificate online using the public key encryption using the TLS communication.

PKI has the root Certificate Authority (CA) which is at the top of the tree which is used to sign all the certificates. A signature of the root has been assigned which is considered as the intermediate CA or subordinate CA. Intermediate CA can verify the certificate signed by the root CA which are distributed among the organization. In our design Master will be assigned as an intermediate CA which will verify the certificate and the PKI design will be implemented in the hierarchical architecture. In this architecture, the PKI connection used the public certificate of the entities communicating and the Intermediate CA to verify the certificate. Every entity or user has a public certificate, private key, and a peer certificate to verify the file.

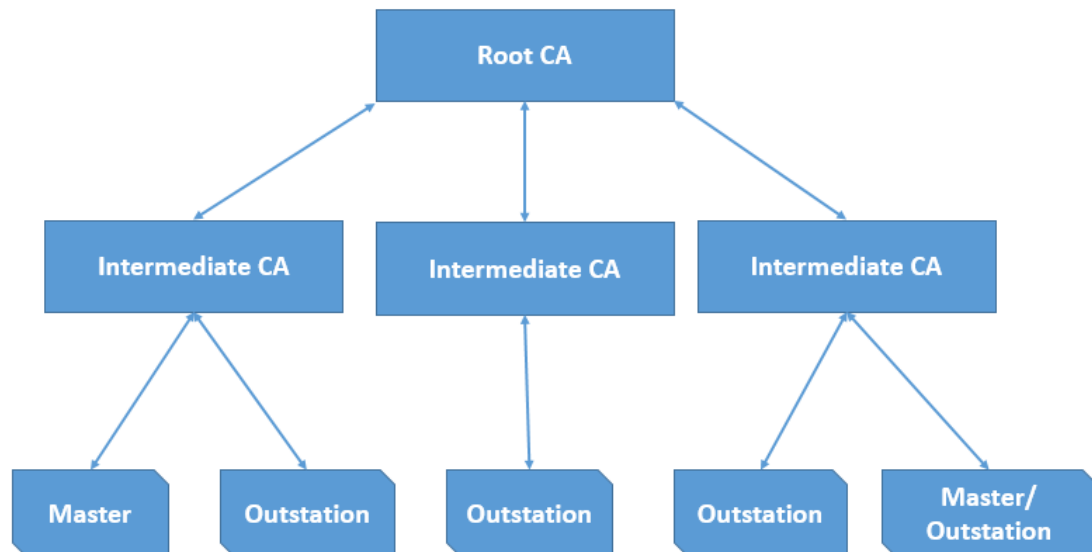


Figure 19. PKI Architecture of Distribution of Certificates

5.2 Configuration for RPI – DNP3

Configuration for the DNP3 such as a change in the source code of the opendnp3 [33] and internet on Wireless Are Network will be explained. From the given source code in Tlsconfig.cpp, the Raspberry PI has a problem recognizing the destination of the certificate and keys were placed in the SSLContect.cpp file. We have also used the socket connection for the verification of the file when an entity sends a public certificate. Key generation and certificate are generated from the Root CA. For the root CA, we have used a third entity on a Kali Linux machine for the certificate storage and revocation list.

5.3 Implementation of PKI

The implementation of Public Key Infrastructure (PKI) with the use of openssl[36] on the DNP3 Secure Authentication Version 5(SAv5) with intermediate CA verification. We have implemented the CA verification using the socket connection with Master or Outstation sending the public certificate for verification. The implementation was designed to run in two models depicting a third part CA using a different system on Internet Protocol (IP). The second model runs on the Hierarchical model where Master, Outstation acts as an intermediate CA to verify the files signed by the root CA. Every entity has a public certificate, private key which is in a PKCS10 format which is translated as PEM.

The First model of the implementation has the Master acting as the Intermediate CA, and Outstation sends its public certificate and master verify the signature of the file. Once verified, a message will be sent verifying the authenticity of the device and details of the certificate signed by the root CA. Later receiving the message, the outstation will send the peer certificate to master and master to outstation for TLS connection will be decrypted by the private keys of entities.

To secure customer substation, added an extra layer of security for the private key to authenticate to other devices by keeping a password to authenticate for the decoding the public certificate. In [26], Aamir Shahzad et. al has cryptographically implemented a solution for secure DNP3 architecture on the application part of the stack but here in this research, we have shown the application stack by implementing the application in real-time on the IOT device.

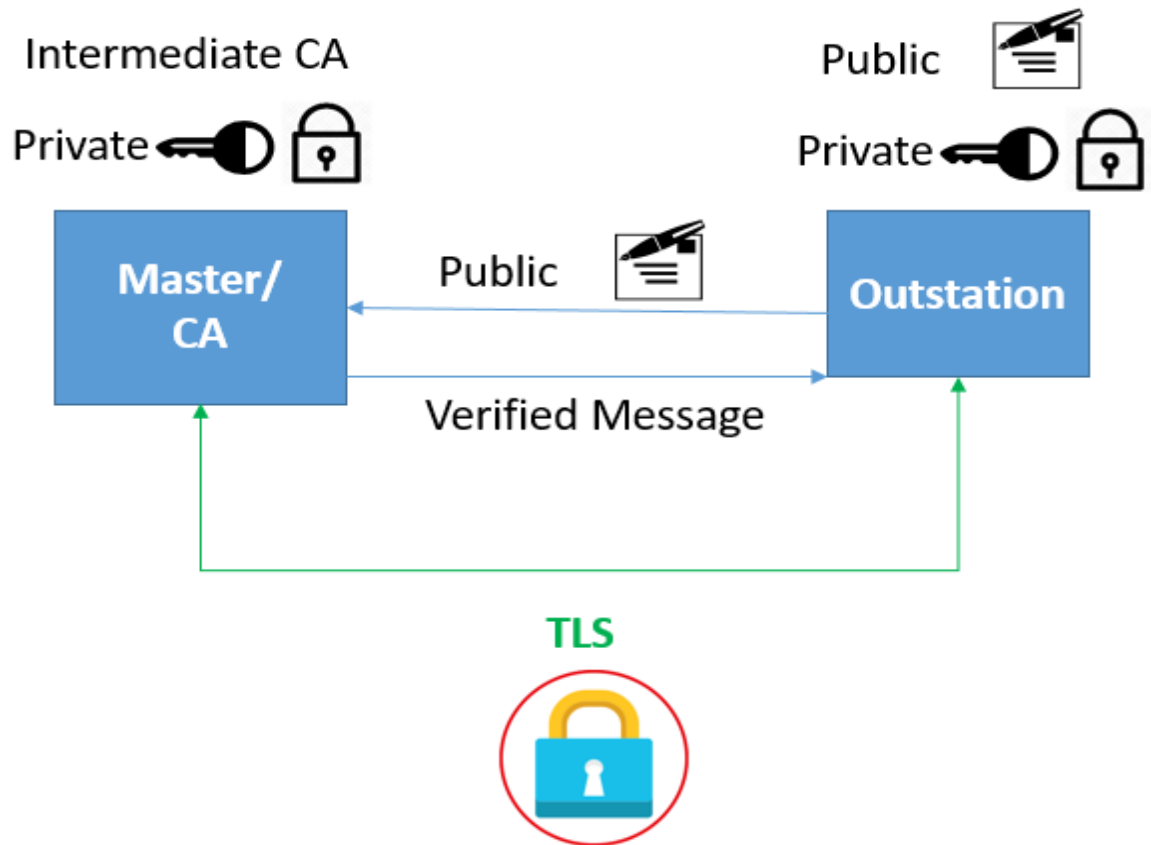


Figure 20. Verification of Outstation Certificate by Master as Intermediate CA

The second model has different intermediate CA on a different machine which is one to one model, and it can verify the master public certificate sent by the outstation. Every entity is communicating here on the Linux operating systems. After the verification message sent to the outstation, the device will send its certificate and both the entities need the private key password to decrypt the certificate. Later, the communication of the devices on the DNP3 SAV5 on the TLS layer with encryption.

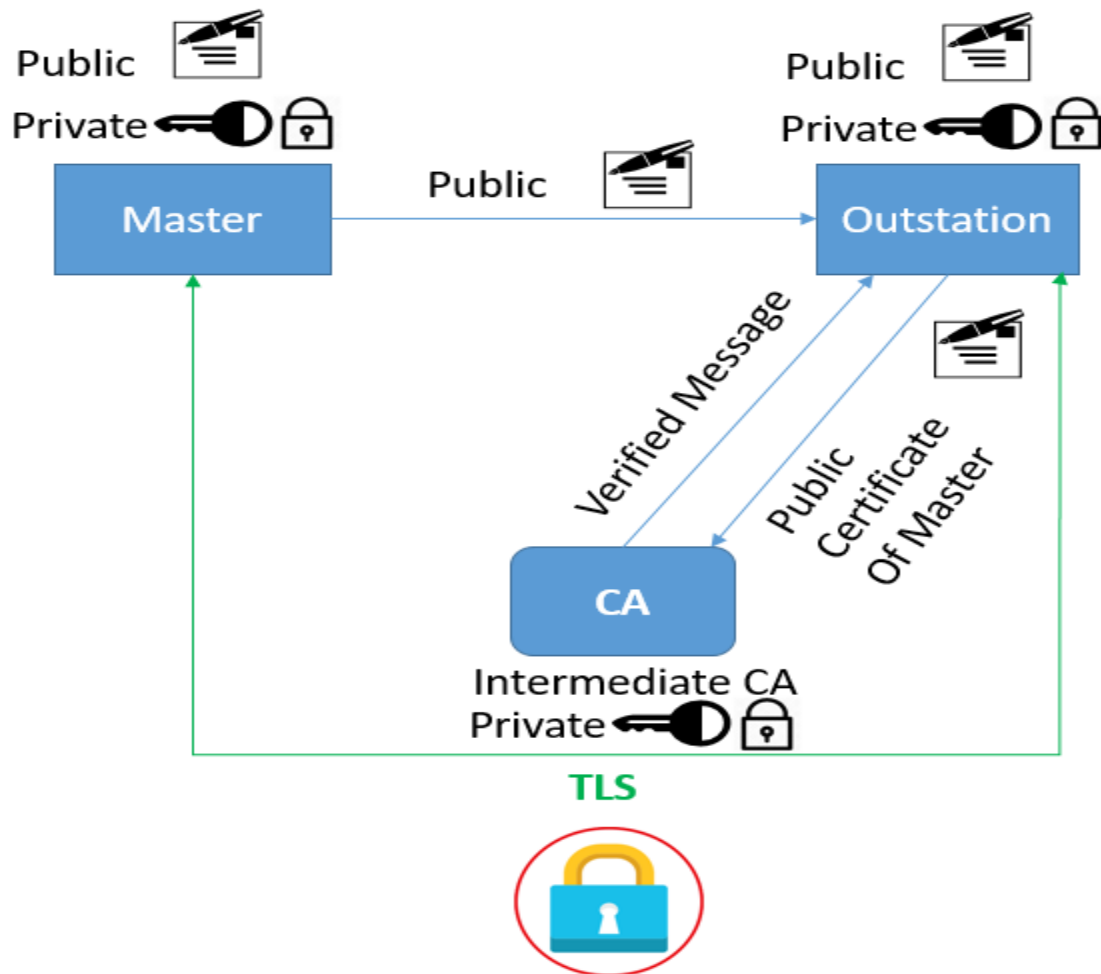


Figure 21. Verification of Outstation Certificate by Master as Intermediate CA

Both models can be used in the hierarchical and one to one architectures in SCADA systems. The model illustrates how our PKI enhanced with intermediate CA will be used in the distribution substation to connect with a smart meter at a customer substation. Physical tampering at the customer will be a security threat to the organization, so we have added an added layer of security by giving password protection of the private key. When a customer sends a public certificate, it will be verified by an intermediate CA, and then a TLS connection will be established. This architecture communicates on the secure layer with authenticity verification and also blocking a security threat at customer substation.

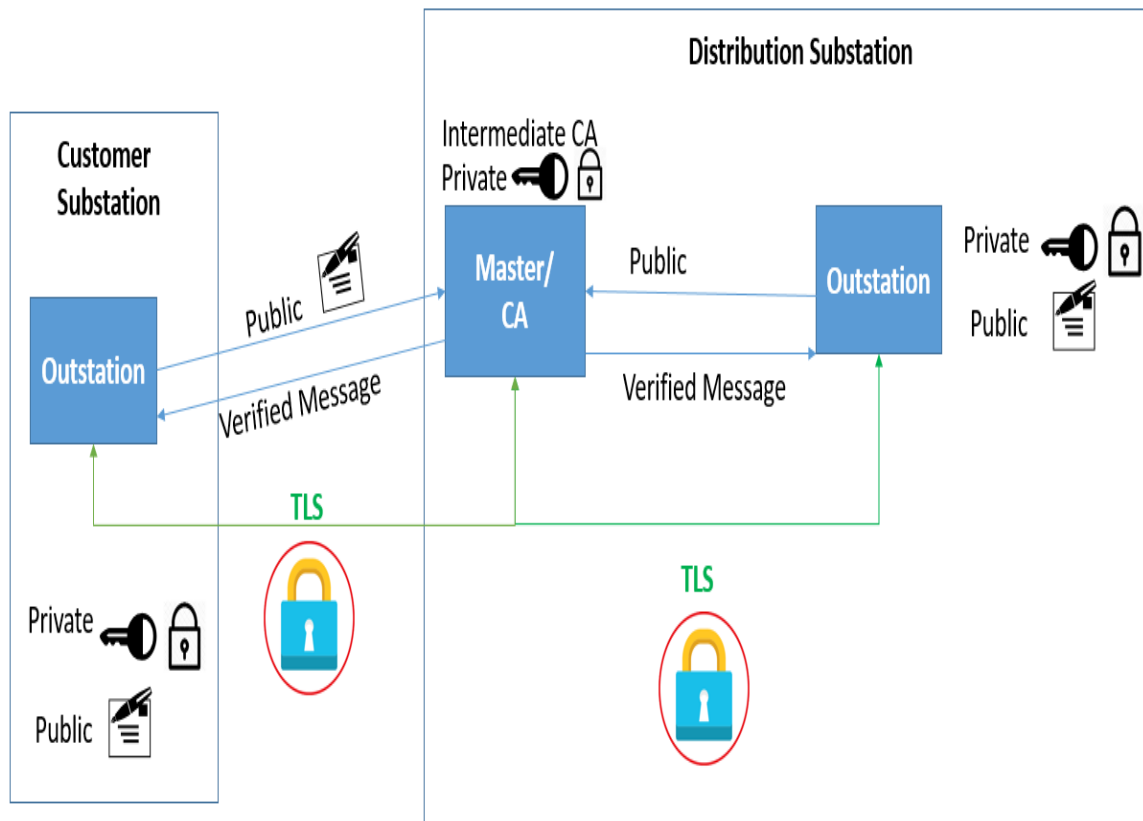
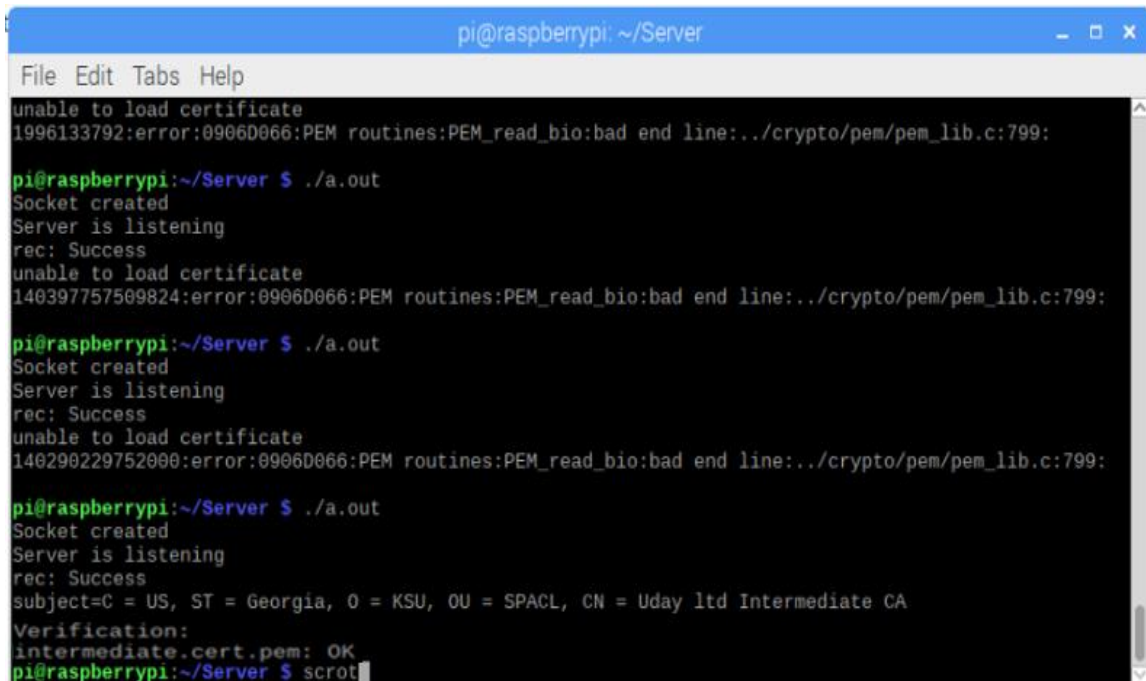


Figure 22. Verification of Outstation Certificate at Customer Substation by Master as Intermediate CA

5.4 Evaluation

In this section, we present the results of simulations run on DNP3 SAV5 using Public Key Infrastructure. Using the Hierarchical model, we built a model where the master communicating with outstations and the verification of the public certificate are shown. Later, the password for the private key will be given for decrypting the public certificate. Connection for TLS communication between master outstation will be shown, and the secure layer communication is captured using the Wireshark application [37]. The captured packet is shown for the TLS communication and encrypted data packet.



```

pi@raspberrypi: ~/Server
File Edit Tabs Help
unable to load certificate
1996133792:error:0906D066:PEM routines:PEM_read_bio:bad end line:../crypto/pem/pem_lib.c:799:

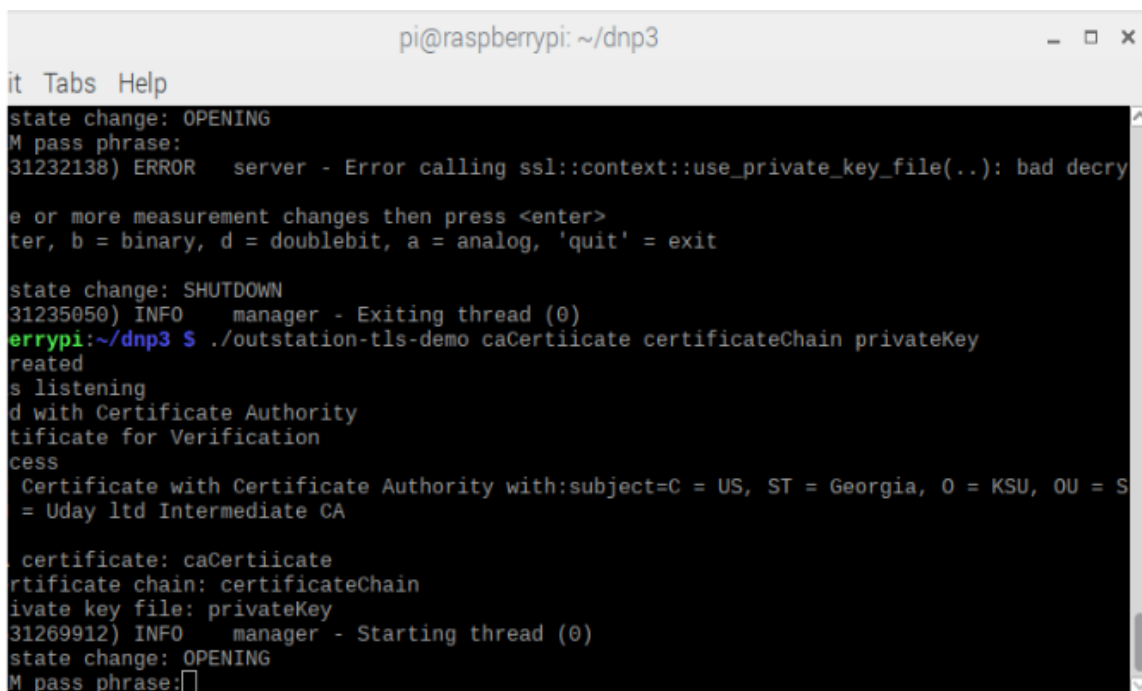
pi@raspberrypi:~/Server $ ./a.out
Socket created
Server is listening
rec: Success
unable to load certificate
140397757509824:error:0906D066:PEM routines:PEM_read_bio:bad end line:../crypto/pem/pem_lib.c:799:

pi@raspberrypi:~/Server $ ./a.out
Socket created
Server is listening
rec: Success
unable to load certificate
140290229752000:error:0906D066:PEM routines:PEM_read_bio:bad end line:../crypto/pem/pem_lib.c:799:

pi@raspberrypi:~/Server $ ./a.out
Socket created
Server is listening
rec: Success
subject=C = US, ST = Georgia, O = KSU, OU = SPACL, CN = Uday ltd Intermediate CA
Verification:
intermediate.cert.pem: OK
pi@raspberrypi:~/Server $ scrot

```

Image 2. Certificate Authority Verification



```

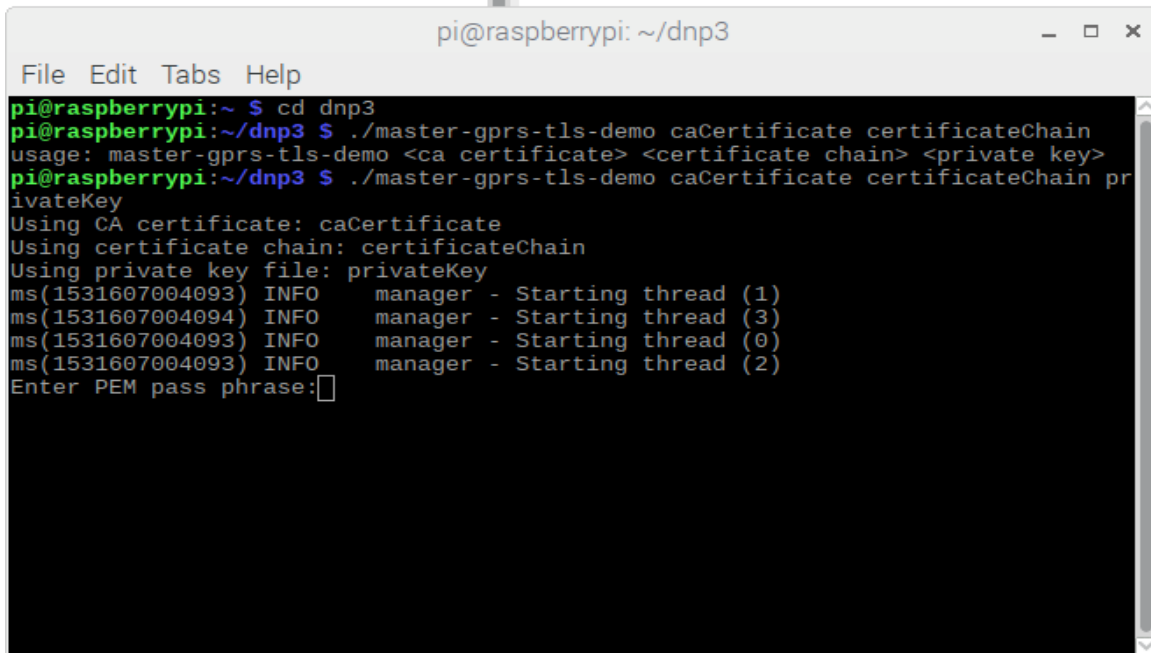
pi@raspberrypi: ~/dnp3
File Edit Tabs Help
state change: OPENING
M pass phrase:
31232138) ERROR    server - Error calling ssl::context::use_private_key_file(..): bad decrypt
or more measurement changes then press <enter>
ter, b = binary, d = doublebit, a = analog, 'quit' = exit

state change: SHUTDOWN
31235050) INFO     manager - Exiting thread (0)
errypi:~/dnp3 $ ./outstation-tls-demo caCertificate certificateChain privateKey
reated
s listening
d with Certificate Authority
tificate for Verification
cess
Certificate with Certificate Authority with:subject=C = US, ST = Georgia, O = KSU, OU = SPACL, CN = Uday ltd Intermediate CA

. certificate: caCertificate
rtificate chain: certificateChain
ivate key file: privateKey
31269912) INFO     manager - Starting thread (0)
state change: OPENING
M pass phrase:

```

Image 3. Outstation Certificate verification, and Private Key Password Request

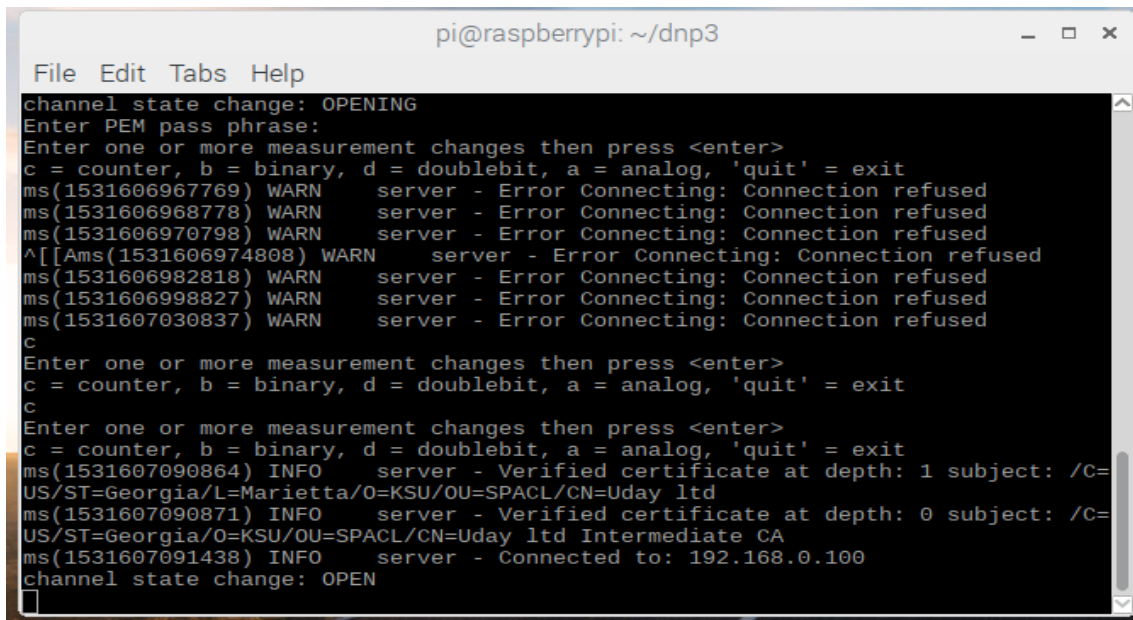


```

pi@raspberrypi: ~/dnp3
File Edit Tabs Help
pi@raspberrypi:~ $ cd dnp3
pi@raspberrypi:~/dnp3 $ ./master-gprs-tls-demo caCertificate certificateChain
usage: master-gprs-tls-demo <ca certificate> <certificate chain> <private key>
pi@raspberrypi:~/dnp3 $ ./master-gprs-tls-demo caCertificate certificateChain privateKey
Using CA certificate: caCertificate
Using certificate chain: certificateChain
Using private key file: privateKey
ms(1531607004093) INFO      manager - Starting thread (1)
ms(1531607004094) INFO      manager - Starting thread (3)
ms(1531607004093) INFO      manager - Starting thread (0)
ms(1531607004093) INFO      manager - Starting thread (2)
Enter PEM pass phrase:

```

Image 4. Master Private Key Password Request



```

pi@raspberrypi: ~/dnp3
File Edit Tabs Help
channel state change: OPENING
Enter PEM pass phrase:
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
ms(1531606967769) WARN      server - Error Connecting: Connection refused
ms(1531606968778) WARN      server - Error Connecting: Connection refused
ms(1531606970798) WARN      server - Error Connecting: Connection refused
^[[Ams(1531606974808) WARN      server - Error Connecting: Connection refused
ms(1531606982818) WARN      server - Error Connecting: Connection refused
ms(1531606998827) WARN      server - Error Connecting: Connection refused
ms(1531607030837) WARN      server - Error Connecting: Connection refused
c
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
c
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
ms(1531607090864) INFO      server - Verified certificate at depth: 1 subject: /C=
US/ST=Georgia/L=Marietta/O=KSU/OU=SPACL/CN=Uday ltd
ms(1531607090871) INFO      server - Verified certificate at depth: 0 subject: /C=
US/ST=Georgia/O=KSU/OU=SPACL/CN=Uday ltd Intermediate CA
ms(1531607091438) INFO      server - Connected to: 192.168.0.100
channel state change: OPEN

```

Image 5. Master certificate decryption and connection with Outstation

```

pi@raspberrypi: ~/dnp3
File Edit Tabs Help
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
ms(1531607090864) INFO      server - Verified certificate at depth: 1 subject: /C=
US/ST=Georgia/L=Marietta/O=KSU/OU=SPACL/CN=Uday ltd
ms(1531607090871) INFO      server - Verified certificate at depth: 0 subject: /C=
US/ST=Georgia/O=KSU/OU=SPACL/CN=Uday ltd Intermediate CA
ms(1531607091438) INFO      server - Connected to: 192.168.0.100
channel state change: OPEN
c
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
c
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
c
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
c
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit
^[[A
Enter one or more measurement changes then press <enter>
c = counter, b = binary, d = doublebit, a = analog, 'quit' = exit

```

Image 6. Outstation Certificate decryption and connection with Master

```

pi@raspberrypi: ~/dnp3
File Edit Tabs Help
Using CA certificate: caCertificate
Using certificate chain: certificateChain
Using private key file: privateKey
ms(1531607004093) INFO      manager - Starting thread (1)
ms(1531607004094) INFO      manager - Starting thread (3)
ms(1531607004093) INFO      manager - Starting thread (0)
ms(1531607004093) INFO      manager - Starting thread (2)
Enter PEM pass phrase:
ms(1531607037380) INFO      server-20001 - Listening on: 0.0.0.0:20001
Enter a command
x - exits program
ms(1531607090860) INFO      server-20001 - Accepted connection from: 192.168.0.104
:32863
ms(1531607091171) INFO      server-20001 - Verified certificate at depth: 1 subjec
t: /C=US/ST=Georgia/L=Marietta/O=KSU/OU=SPACL/CN=Uday ltd
ms(1531607091177) INFO      server-20001 - Verified certificate at depth: 0 subjec
t: /C=US/ST=Georgia/O=KSU/OU=SPACL/CN=Uday ltd Intermediate CA
ms(1531607091450) <-LL-- session-0 - Function: PRI_UNCONFIRMED_USER_DATA Dest:
1 Source: 10 Length: 10
ms(1531607091451) <-TL-- session-0 - FIR: 1 FIN: 1 SEQ: 0 LEN: 4
ms(1531607091451) <-AL-- session-0 - FIR: 1 FIN: 1 CON: 1 UNS: 1 SEQ: 0 FUNC: U
NSOLICITED_RESPONSE IIN: [0x82, 0x00]
ms(1531607091451) --AL-> session-0 - D0 00
ms(1531607091451) --AL-> session-0 - FIR: 1 FIN: 1 CON: 0 UNS: 1 SEQ: 0 FUNC: C

```

Image 7. Master receiving DNP3 messages

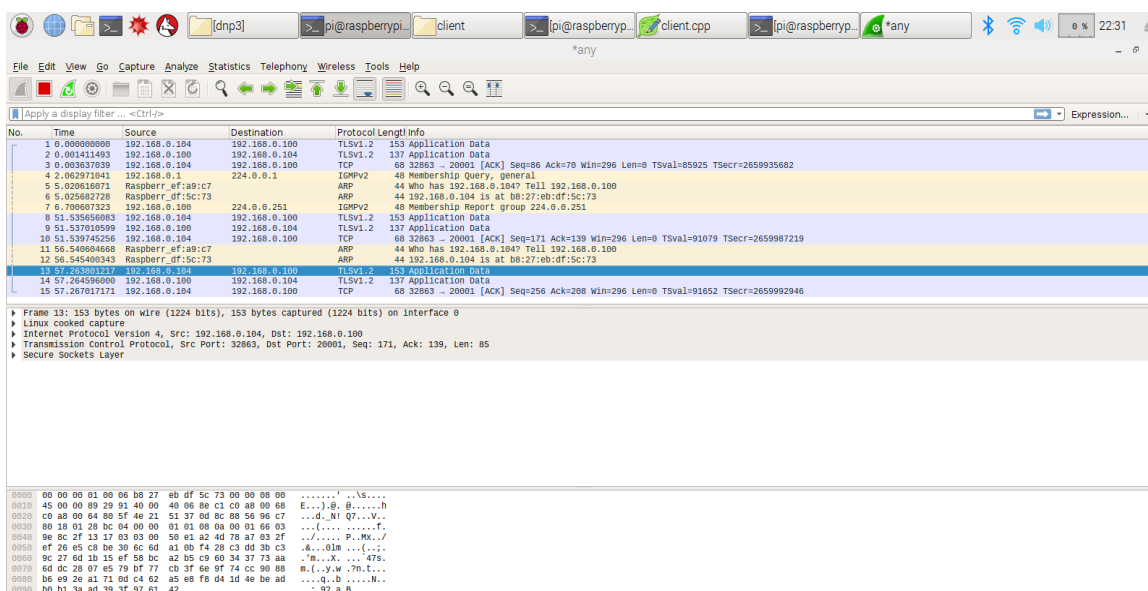


Image 8. Wireshark Capture

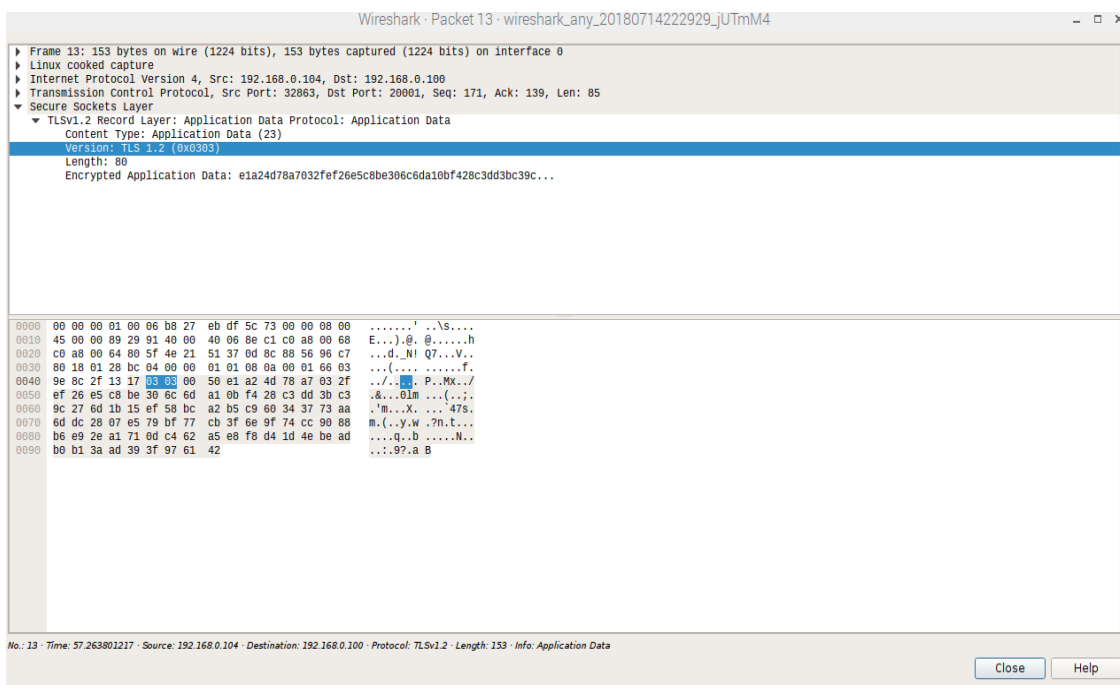


Image 9. TLS version 1.2 Encrypted Data Packet

CHAPTER VI

CONCLUSION

This research presented the secure architecture of DNP3 SAv5 in a SCADA system test environment to be used for the secure communication. The developed test environment was to illustrate the secure communication with the help of Public Key Infrastructure (PKI) in the SCADA systems. We have also explained how PKI plays a significant part in digitally signing a certificate and securely transferring messages between the SCADA systems. The test illustrated the TLS communication between a distribution and customer substation to defend the attacks such tampering or Man in the middle cyber-attacks. We have added a layer of security such as intermediate CA verification and Private Key file password. This will allow the authenticity of the entity is verified and also block the physical tampering of the device.

Results show the feasibility of protecting the cyber-attacks from various research done on the vulnerability assessment on DNP3 communication in an Industrial control system environment. Our next stage is to enhance the architecture by adding detection models for the different network system attacks.

CHAPTER VII

REFERENCES

- [1] A. Leonardi, K. Mathioudakis, A. Wiesmaier, and F. Zeiger, “Towards the Smart Grid: Substation Automation Architecture and Technologies,” *Advances in Electrical Engineering*, Volume 2014, Article ID 896296, 13 pages, 20 August 2014.
- [2] Byron Flynn, “Secure Substation Automation for Operations & Maintenance,” GE Energy.
- [3] Ihab Darwish, Obinna Igbe, Orhan Celebi, Tarek Saadawi, Joseph Soryal, “Smart Grid DNP3 Vulnerability Analysis and Experimentation”, 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, 07 January 2016.
- [4] Cas Cremers, Martin Dehnel-Wild, Kevin Milner, “Secure Authentication in the Grid: A Formal Analysis of DNP3: SAv5”, Published 2017 in ESORICS.
- [5] Raphael Amoah, Seyit Camtepe, Ernst Foo, “Securing Dnp3 Broadcast Communications In SCADA Systems”, *IEE Transactions On Industrial Informatics*, Vol. 12, No. 4, August 2016.

- [6] Carlos Lopez, Arman Sargolzaei, Hugo Santana, Carlos Huerta, “Smart Grid Cyber Security: An Overview of Threats and Countermeasures,” *Journal of Power and Energy Engineering*, July 2015.
- [7] Dongsoo Lee, HakJu Kim, Kwangjo Kim, Paudl D. Yoo, “Simulated Attack on DNP3 Protocol in SCADA System”, *The Institute of Electronics, Information and Communication Engineers*, 2014.
- [8] Samuel East, Jonathan Butts, Mauricio Papa, and Sujeet Sheno, “A Taxonomy of attacks on the DNP3 protocol”, *Critical Infrastructure Protection III, IFIP Advances in Information and Communication Technology*, 2009.
- [9] J. Adam Crain, Sergey Bratus, “Bolt-On Security Extensions for Industrial Control System Protocol System Protocols: A Case Study of DNP3 SAv5”, *IEEE Computer and Reliability Societies*, June 2015.
- [10] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo, “Cyber-Physical Systems Security – A Survey”, 17 January 2017.
- [11] Marcio Andrey Teixeira, Tara Salman, Mohammed Samaka, Raj Jain, and Nader Moslem, “SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach”, *Future Internet* 2018.
- [12] Prof. More V.N. “Authentication and Authorization Models,” *International Journal of Computer Science and Security (IJCSS)*, 2011.

- [13] Richard Duncan, “An Overview of Different Authentication Methods and Protocols,” SANS Institute InfoSec Reading Room, 2002.
- [14] Jacques Benoit, “An Introduction to Cryptography as Applied to the Smart Grid,” Cooper Power Systems.
- [15] Chih-Che Sun, Adam Hahn, Chen-Ching Liu, “Cybersecurity of a power grid: State-of-art,” Electrical Power and Energy Systems, 04 January 2018.
- [16] Eric J. Byres, Matthew Franz, and Darrin Miller, “The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems.”
- [17] Alcides Ortega, Ailton A. Shinoda, Christiane M. Schweitzer, Fabrizio Granelli, Aleciana V. Ortega, and Fabiola Bonvecchio, “Performance Evaluation of the DNP3 Protocol for Smart Grid Applications over IEEE 802.3/802.11 Networks and Heterogeneous Traffic”.
- [18] Ihab Darwish, Obinna Igbe and Tarek Saadawi, “Vulnerability Assessment and Experimentation of Smart Grid DNP3”, Journal of Cyber Security, Vol. 5, 29 June 2016.
- [19] Rajendra Kumar Pandey, Mohit Misra, “Cyber Security Threats – Cyber Grid Infrastructure,” 2016.

- [20] Adnan Anwar, Abdun Naser Mahmood, “Cyber Security of Smart Grid Infrastructure,” January 2014.
- [21] Wenye Wang, Zhuo Lu, “Cyber Security in the Smart Grid: Survey and Challenges,” November 1, 2012.
- [22] Gordon Clarke, Deon Reynders, and Edwin Wright, “Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems”, 2004.
- [23] Dong Jin, Guanhua Yan, “An Event Buffer Flooding Attack in DNP3 Controlled SCADA Systems”, 2011 Winter Simulation Conference.
- [24] James H. Graham, Sandip C. Patel, “Intelligent Systems Research Laboratory,” September 2004.
- [25] Neetesh Saxena, and Santiago Grijalva, “Dynamic Secrets and Secret Keys Based Scheme for Securing Last Mile Smart Grid Wireless Communication,” 2016.
- [26] Aamir Shahzad, Malrey Lee, Suntae Kim, Kangmin Kim, Jae-Young Choi, Younghwa Cho, and Keun-Kwang Lee, “Design and Development of Layered Security: Future Enhancements and Directions in Transmission,” MDPI Journal 2016.
- [27] “A DNP3 Protocol Primer”, Ken Curtis, 20 March 2005

- [28] “IEEE Recommended Practice for Data Communications between Remote Terminal Units and Intelligent Electronic Devices in a Substation,” IEEE-SA Standards Bard, 1 January 2001.
- [29] “1815.1-2015 – Draft Standard for Exchanging Information Between Networks Implementing IEC 61850 and IEEE Std 1815 (Distributed Network Protocol – DNP3) 2016.
- [30] “1815-2012 - IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3)”, IEEE Std 1815-2010.
- [31] “Why IEEE 1815 (DNP3) Secure Authentication?”
<https://www.dnp.org>.
- [32] “Further Information Regarding the Release of DNP3 Secure Authentication Version 5 (SAv5)”, <https://www.dnp.org>, 1 December 2011.
- [33] Opendnp3 – Automatak - <https://github.com/automatak/dnp3>
- [34] GCC Compiler - <https://gcc.gnu.org/>
- [35] CMake Library - <https://cmake.org/>
- [36] OpenSSL - <https://www.openssl.org/>
- [37] “Wireshark – Network Protocol Analyzer”,
<https://www.wireshark.org/>

CHAPTER VIII

SOURCE CODE

1.1 OUTSTATION CODE

```

#include <asiodnp3/DNP3Manager.h>
#include <asiodnp3/PrintingSOEHandler.h>
#include <asiodnp3/PrintingChannelListener.h>
#include <asiodnp3/ConsoleLogger.h>
#include <asiodnp3/UpdateBuilder.h>
#include <asiopal/UTCTimeSource.h>
#include <opendnp3/outstation/SimpleCommandHandler.h>
#include <opendnp3/outstation/Database.h>
#include <opendnp3/LogLevels.h>
#include <string>
#include <thread>
#include <iostream>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<netdb.h>
#include <sys/uio.h>
#include <unistd.h>
#include <fcntl.h>
#include<stdio.h>
#include<string.h>
#include <sstream>
#include<sys/syscall.h>

using namespace std;
using namespace opendnp3;
using namespace openpal;
using namespace asiopal;
using namespace asiodnp3;

#define PORT 5555
#define BACKLOG 5

void ConfigureDatabase(DatabaseConfig& config)
{
    // example of configuring analog index 0 for Class2
    // with floating point variations by default
    config.analog[0].clazz = PointClass::Class2;
    config.analog[0].svariation =

```



```

StaticAnalogVariation::Group30Var5;
    config.analog[0].evariation =
EventAnalogVariation::Group32Var7;
}

struct State
{
    uint32_t count = 0;
    double value = 0;
    bool binary = false;
    DoubleBit dbit = DoubleBit::DETERMINED_OFF;
};

void AddUpdates(UpdateBuilder& builder, State& state,
const std::string& arguments);

int main(int argc, char* argv[])
{
    int socket1, socket_accept;
    int bnd,lstn;
    char buf[2048]={ ' '};
    char data[2048] = { ' '};
    struct sockaddr_in server,client;

    //create socket
    socket1=socket(AF_INET,SOCK_STREAM,0);
    if(socket1<0)
    {
        cout<<"Error creating socket\n";
        return 0;
    }
    cout<<"Socket created\n";

    server.sin_family=AF_INET;
    server.sin_port=htons(PORT);
    server.sin_addr.s_addr=INADDR_ANY;

    //Bind socket
    bnd=bind(socket1,(struct sockaddr
*) &server,sizeof(server));
    if(bnd== -1)
    {
        cout<<"Error binding\n";
        return 0;
    }
}

```

```

//listen for socket
lstn=listen(socket1, BACKLOG);
if(lstn==-1)
{
    cout<<"Error listening\n";
    return 0;
}

cout<<"Server is listening\n";
socklen_t len=sizeof(client);
socket_accept=accept(fdl, (struct
sockaddr*)&client, &len);

if(socket_accept==0)
{
    cout<<"Error accepting\n";
    return 0;
}
cout<<"Connected with Certificate Authority"<<endl;
//send Certificate file
int from;
from=open("Uday.cert.pem", O_RDONLY);

if(from<0)
{
    cout<<"Error opening file\n";
    return 0;
}
int n=1;
int s;
while((n=read(from, buf, sizeof(buf)))!=0)
{
    //s=send(fd2, buf, sizeof(buf), 0);
    s=write(socket_accept, buf, n);

    if(s<0)
    {
        cout<<"error sending\n"; return 0;
    }
    break;
}
cout<<"Sent Certificate for Verification"<<endl;
int rec;
while(1)
{
    rec = recv(socket_accept, data, sizeof(data), 0);
    perror("rec");
}

```

```

        if(rec<0)
        {
            cout<<"Error Receiving\n";

        }
        cout<< "Verified Certificate with Certificate
Authority with:"<<data <<endl;
        break;
    }

    if (argc != 4)
    {
        std::cout << "usage: master-gprs-tls-demo <ca
certificate> <certificate chain> <private key>" <<
std::endl;
        return -1;
    }

    std::string caCertificate(argv[1]);
    std::string certificateChain(argv[2]);
    std::string privateKey(argv[3]);

    std::cout << "Using CA certificate: " << caCertificate
<< std::endl;
    std::cout << "Using certificate chain: " <<
certificateChain << std::endl;
    std::cout << "Using private key file: " << privateKey
<< std::endl;

    // Specify what log levels to use. NORMAL is warning
and above
    // You can add all the comms logging by uncommenting
below.
    const uint32_t FILTERS = levels::NORMAL; // |
levels::ALL_COMMS;

    // This is the main point of interaction with the
stack
    // Allocate a single thread to the pool since this is
a single outstation
    DNP3Manager manager(1, ConsoleLogger::Create());

    std::error_code ec;

    // Create a TCP server (listener)
    auto channel = manager.AddTLSClient(
        "server",

```

```

        FILTERS,
        ChannelRetry::Default(),
        "192.168.0.100",
        "0.0.0.0",
        20001,
        TLSConfig(
            caCertificate,
            certificateChain,
            privateKey,
            2
        ),
        PrintingChannelListener::Create(),
        ec
    );

    if (ec)
    {
        std::cout << "Unable to create tls server: " <<
ec.message() << std::endl;
        return ec.value();
    }

    // The main object for a outstation. The defaults are
    useable,
    // but understanding the options are important.
    OutstationStackConfig
    stackConfig(DatabaseSizes::AllTypes(10));

    // specify the maximum size of the event buffers
    stackConfig.outstation.eventBufferConfig =
    EventBufferConfig::AllTypes(10);

    // you can override an default outstation parameters
    here
    // in this example, we've enabled the oustation to use
    unsolicited reporting
    // if the master enables it
    stackConfig.outstation.params.allowUnsolicited = true;

    // You can override the default link layer settings
    here
    // in this example we've changed the default link
    layer addressing
    stackConfig.link.LocalAddr = 10;
    stackConfig.link.RemoteAddr = 1;

    // You can optionally change the default reporting

```

```

variations or class assignment prior to enabling the
outstation
    ConfigureDatabase(stackConfig.dbConfig);

    // Create a new outstation with a log level, command
    handler, and
    // config info this returns a thread-safe interface
    used for
    // updating the outstation's database.
    auto outstation = channel->AddOutstation("outstation",
    SuccessCommandHandler::Create(),
    DefaultOutstationApplication::Create(), stackConfig);

    // Enable the outstation and start communications
    outstation->Enable();

    // variables used in example loop
    string input;
    State state;

    while (true)
    {
        std::cout << "Enter one or more measurement changes
        then press <enter>" << std::endl;
        std::cout << "c = counter, b = binary, d =
        doublebit, a = analog, 'quit' = exit" << std::endl;
        std::cin >> input;

        if (input == "quit") return 0;
        else
        {
            UpdateBuilder builder;
            AddUpdates(builder, state, input);
            outstation->Apply(builder.Build());
        }
    }

    return 0;
}

void AddUpdates(UpdateBuilder& builder, State& state,
const std::string& arguments)
{
    for (const char& c : arguments)
    {
        switch (c)
        {

```

```

        case('c'):
        {
            builder.Update(Counter(state.count), 0);
            ++state.count;
            break;
        }
        case('a'):
        {
            builder.Update(Analog(state.value), 0);
            state.value += 1;
            break;
        }
        case('b'):
        {
            builder.Update(Binary(state.binary), 0);
            state.binary = !state.binary;
            break;
        }
        case('d'):
        {
            builder.Update(DoubleBitBinary(state.dbit),
0);
            state.dbit = (state.dbit ==
DoubleBit::DETERMINED_OFF) ? DoubleBit::DETERMINED_ON :
DoubleBit::DETERMINED_OFF;
            break;
        }
        default:
            break;
    }
}
}

```